Extending Battery System Operation via Adaptive Reconfiguration

LIANG HE, University of Colorado Denver
LINGHE KONG, Shanghai Jiaotong University
YU GU, Visa Inc.
CONG LIU, University of Texas at Dallas
TIAN HE, University of Minnesota
KANG G. SHIN, University of Michigan

11

Large-scale battery packs are commonly used in applications such as electric vehicles (EVs) and smart grids. Traditionally, to provide stable voltage to the loads, voltage regulators are used to convert battery packs' output voltage to those of the loads' required levels, causing power loss especially when the difference between the supplied and required voltages is large or when the load is light. In this article, we address this issue via a reconfiguration framework for the battery system. By abstracting the battery system as a cell graph, we develop an adaptive reconfiguration algorithm to identify the desired system configurations based on real-time load requirements. Our design is evaluated via both prototype-based experiments, EV driving trace-based emulations, and large-scale simulations. The results demonstrate an extended system operation time of up to 5×, especially when facing severe cell imbalance.

CCS Concepts: • **Computer systems organization → Embedded software**;

Additional Key Words and Phrases: Reconfigurable battery packs, adaptive reconfiguration, rate capacity effect, cell imbalance, voltage regulation

ACM Reference format:
Liang He, Linghe Kong, Yu Gu, Cong Liu, Tian He, and Kang G. Shin. 2019. Extending Battery System Operation via Adaptive Reconfiguration. *ACM Trans. Sen. Netw.* 15, 1, Article 11 (January 2019), 21 pages.
https://doi.org/10.1145/3284556

An early version of this work has been published at IEEE RTSS'13 (CPS track) (He et al. 2013).
The work reported in this article was supported in part by NSF under Grants CNS-1446117, CNS-1503590, CNS 1527727, CNS-1739577, CNS-1750263, and NSFC under 61629302.
Authors' addresses: L. He, Department of Computer Science and Engineering, University of Colorado Denver, Lawrence Street Center, 1380 Lawrence Street, Suite 800, Denver, Colorado 80204; L. Kong (corresponding author), Computer Science Department, Shanghai Jiaotong University, Dongchuan Rd., Shanghai, China, 200240; email: linghe.kong@sjtu.edu.cn; Y. Gu, Visa Inc., 12301 Research Blvd, Austin, TX, 78759; C. Liu, Department of Computer Science, University of Texas at Dallas, Richardson, Texas, 75080; T. He, Department of Computer Science and Engineering, University of Minnesota, 200 Union Street SE Minneapolis, MN 55455; K. G. Shin, Department of Electrical Engineering and Computer Science, University of Michigan, 2260 Hayward St., Ann Arbor, MI 48109-2121.

1550-4859/2019/01-ART11 $15.00
https://doi.org/10.1145/3284556

1 INTRODUCTION

Large-scale battery systems are now widely used in electric vehicles (EVs) and energy storage in grids. For many of these applications, the load required voltage is dynamic and could change significantly over time (Kim et al. 2011b)—e.g., EV's operating voltage may vary from 300 to 380V. Such dynamic loads make the management of large battery systems more critical and challenging, which has been attracting increasing research attention (Raychev et al. 2011; Kim et al. 2011b, 2012; Vatanparvar and Faruque 2016).

A traditional approach to handle such dynamic voltage requirements is to use voltage regulators to convert the battery pack supplied voltage to the level that is required by the loads. Unfortunately, the power loss on voltage regulators becomes significant when (i) the difference between the supplied and required voltages is large (Kim et al. 2012), or (ii) the load is light (Visairo and Kumar 2008).

Dynamically adjusting the battery pack configuration (i.e., the connections among cells inside the battery pack) based on the real-time load requirements and cell states, referred to as the *adaptive system reconfiguration*, is an alternative approach to handle the mismatch between the voltage supply and requirement (Ci et al. 2012a, 2016). The adaptive reconfiguration not only mitigates the power loss of traditional regulator-based solutions, but also increases the system robustness in that failed cells can be bypassed without significantly degrading system performance (Kim et al. 2011a). Much research has been conducted to design battery systems that offer higher reconfigurability with fewer supplementary electronic components such as connectors and switches (Ci et al. 2012a, 2012b; Alahmad et al. 2008; Raychev et al. 2011).

Besides offering reconfigurability, another open challenge is to determine the proper system configuration based on real-time load requirements. In this article, we address the question that *with given reconfigurability of a battery system, how to adaptively identify proper system configurations based on real-time load requirements to achieve extended system operation.*

Optimizing battery pack performance is challenging because of the complex non-linear cell properties, making the traditional analytical modeling approach either lack accuracy or suffer from high complexity (Ceder et al. 2002; Chou et al. 2003; Cloth et al. 2007; Rong and Pedram 2006; Rakhmatov and Vrudhula 2003; Moura et al. 2013). We circumvent these complex cell properties from engineering via two empirically observed design principles—matching the supplied and required voltages and reducing the discharge current of individual cells—based on which an adaptive reconfiguration algorithm is proposed. This article makes three major contributions.

- —We propose a generic graph representation of reconfigurable battery packs, facilitating the identification of proper system configuration.
- —With the graph representation, we transform the problem of identifying proper system configuration to a *path selection* problem in graphs. We first prove the problem is *NP-Hard*. We then propose a practically feasible solution based on two empirical observations, which returns a promising system configuration via an integration of Depth-First-Search (DFS)-with-pruning and 0-1 programming.
- —We extensively evaluate the design with empirical experiments on a prototype battery system, EV driving trace-based emulations, and large-scale simulations. The evaluation results show that, depending on the battery states, the proposed adaptive reconfiguration achieves up to 5× improvements in system operation time.

The article is organized as follows. The preliminaries and design principles are presented in Sections 2 and 3. The graph representation of battery packs is presented in Section 4. The proposed adaptive reconfiguration is presented in Section 5. The evaluation results are presented in

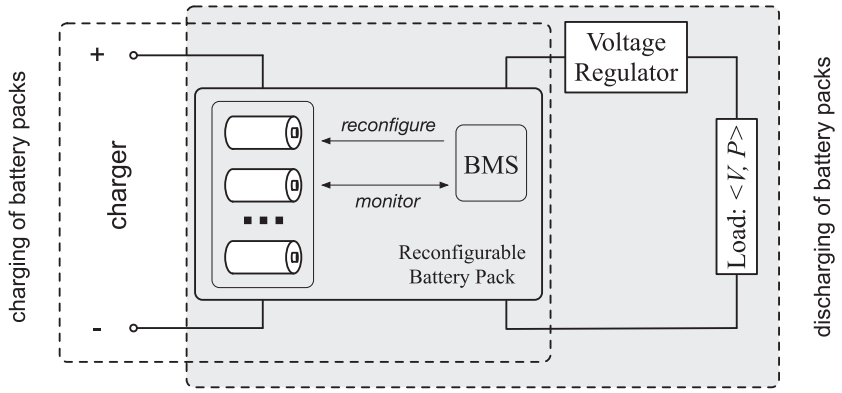


Fig. 1. System model.

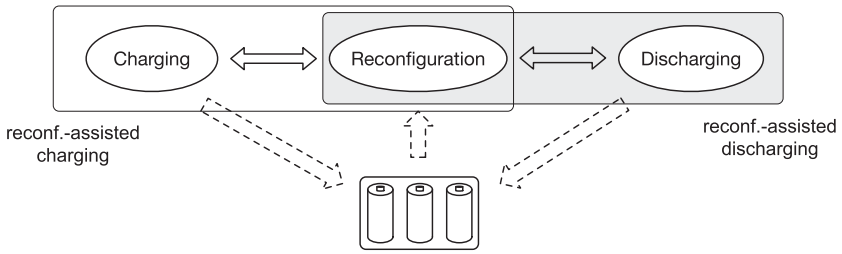


Fig. 2. Transition diagram of system states.

Section 6. Section 7 discusses practical issues and future investigations. Section 8 reviews the literature. Section 9 concludes the article.

2 PRELIMINARIES

Here we introduce the system model and basic background on cell configuration.

2.1 System Model

We consider the discharging of the battery system shown in Figure 1. The battery management system (BMS) monitors a reconfigurable battery pack consisting of N cells and determines its proper configuration to support the load. For notation convenience, let V and P denote the load required voltage and power, respectively, which may change over time. Denote the voltage of the ith cell at the decision time by $v_i \in [v_c, v_f]$, where v_c and v_f are the cutoff voltage[1] and full charge voltage, respectively. Because the cell voltage decreases when powering the load, the battery pack supplied voltage may deviate from that required by the load. A traditional solution is to use a voltage regulator between the battery pack and the load to ensure the required voltage V is supplied to the load (Zeljkovic et al. 2014). This, however, is at the cost of additional power loss on the voltage regulator. Our objective is to design an efficient reconfiguration algorithm for the BMS to identify the battery pack configuration to support the load, achieving an extended system operation.

Figure 2 shows the state transition diagram of the battery pack. The battery states are monitored during both charging and discharging, serving as the input when reconfiguring the battery pack. Besides improving the battery pack discharging as the focus of this article, reconfiguration could also be explored to improve the charging of battery packs as in He et al. (2016a).

[1]The voltage level defines the empty state of battery cells.

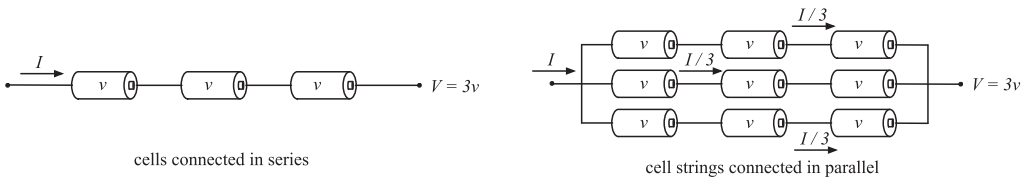


Fig. 3. Series and parallel cell connection.

2.2 Series and Parallel Connection of Cells

Cells in a battery pack can be connected in series or in parallel, as shown in Figure 3. The supplied voltage of series connected cells, i.e., a *cell string*, is the sum of their individual voltages, and these cells share the same discharge current. When further connecting multiple such cell strings in parallel, the overall supplied voltage does not change but the discharge currents on each of these strings will be proportionally reduced (e.g., the discharge current is reduced to $I/3$ when three strings are parallelly connected as in Figure 3.

3 DESIGN PRINCIPLES AND CHALLENGES

Intuitively, an extended system operation can be achieved by (i) reducing the power loss on voltage regulator and (ii) increasing the capacity delivery of battery pack. This section presents our design principles to achieve such objectives and their corresponding challenges.

3.1 Design Principles

3.1.1 Matching Voltages to Reduce Power Loss. The power loss on regulators increases with larger difference between the supplied and required voltages. For example, the power loss of widely used linear regulators can be calculated as

$$P_{\text{loss}} = (V_{\text{sup.}} - V_{\text{req.}}) \cdot I_{\text{load}}. \tag{1}$$

The fact that a larger difference between the supplied and required voltages increases the power loss holds even for more advanced low drop-out regulators. The efficiency of linear voltage regulator is reported to be only about 40%, and could reach as low as 14% (Engineering 2014). Therefore, to reduce the power loss on regulators, it is key to *match the supplied voltage with the load required voltage.*

For reconfigurable battery pack, its supplied voltage is determined by the series connection of its cells. Thus, to match the supplied and required voltages, we need to determine how many cells in the pack should be connected in series.

3.1.2 Reducing Cells' Discharge Current to Enhance Capacity Delivery. Cell's capacity delivery hinges crucially on its discharge current—more capacity can be delivered if the cell is discharged with smaller current, which is commonly known as the *rate-capacity effect.* To clearly demonstrate the rate-capacity effect, Figure 4 plots our measurement when discharging a 2,800mAh Li-ion cell (Battery 2018) with currents of 200–1000mA, with a NEWARE Battery Testing System (Neware 2018). The delivered capacity when discharging with 1,000mA current is about 200mAh less than that delivered with a discharge current of 200mA.

Connecting cell strings in parallel proportionally reduces the discharge current of individual cells. This way, we should (i) identify as many cell strings as possible, where each of them can efficiently support the load by providing a voltage close to the load required level; and (ii) connect these strings in parallel to reduce the discharge current of individual cells.

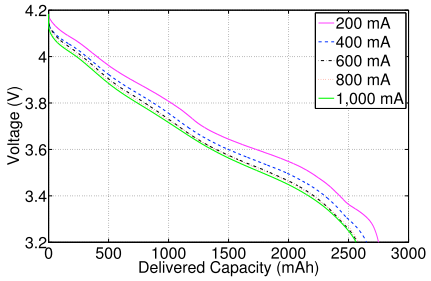


Fig. 4. Cells' capacity delivery is enhanced with smaller discharge currents.

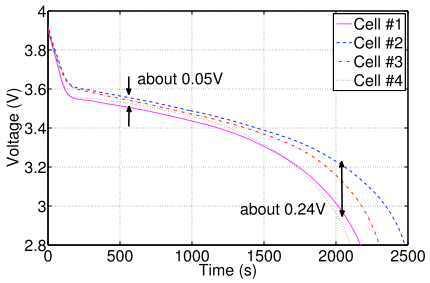


Fig. 5. The voltage traces of four cells in a battery pack, demonstrating clear heterogeneity.

3.2 Design Challenges

The challenges when implementing the above two principles rise from two facts: the *limited reconfigurability of the battery pack* and the *cell imbalance in the pack*.

3.2.1 Limited Reconfigurability of Battery Pack. Complementary components such as switches are needed to make the battery pack reconfigurable, which on the other hand increase system complexity. For example, two switches are used per cell in the reconfigurable battery pack designed in Kim et al. (2011b), while four switches per cell are required in Alahmad et al. (2008). As a result, only limited reconfigurability is offered in existing reconfigurable battery packs due to the concerns on system complexity—not any subset of cells can be connected in series to form a cell string.

3.2.2 Cell Imbalance in Battery Pack. Due to uncontrollable factors such as operation environment and manufacturing, the strength of cells in a battery pack will be different over time and usage, which is commonly referred to as the *cell imbalance issue*. A consequence of cell imbalance is that cells in the pack may have different voltages. To clearly demonstrate the cell voltage heterogeneity, we disassembled a Li-ion laptop battery pack and discharged its four cells individually. The voltage curves of the four cells during discharging are plotted in Figure 5, where obvious difference in cell voltages can be observed, e.g., as large as 0.24V. This voltage heterogeneity indicates that determining *how many* cells should be connected in series is not enough to ensure the matching between the supplied and required voltages, and we also need to identify *which* cells should be selected to form a string.

4 BATTERY PACK GRAPH

We mitigate the above two challenges with a novel weighted graph representation of reconfigurable battery packs—the graph structure (i.e., vertices and directed edges) helps to describe the battery pack's (limited) reconfigurability, and the vertex weight captures the cell voltage heterogeneity. Specifically, given a battery pack and the cell voltages at decision time, we construct a corresponding weighted and directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{W})$ in the way that

(1) the vertex set $\mathcal{V}$ represents the cells in the pack, denoted as $\mathcal{V} = \{n_1, n_2, \ldots, n_N\}$;
(2) the directed edge set $\mathcal{E}$ represents the reconfigurability of the pack, i.e., how the cells can be connected. A directed edge $n_i \rightarrow n_j \in \mathcal{E}$ if and only if the discharge current can flow from n_i to n_j without passing through any other cells;
(3) the weight of each vertex is the voltage of the corresponding cell at decision time: $\forall w_i \in \mathcal{W}, w_i = v_i (i = 1, 2, \ldots, N)$.

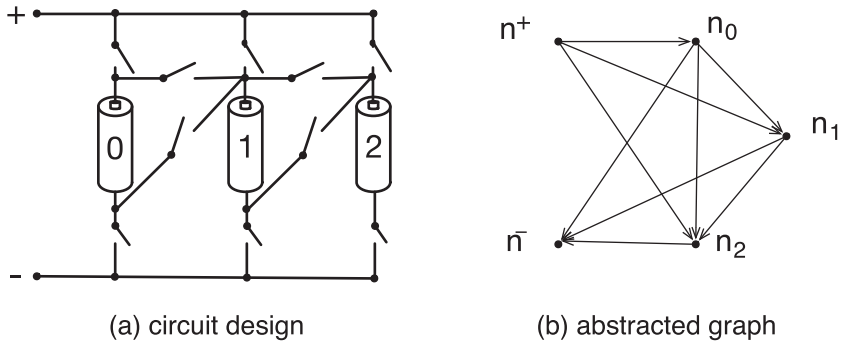


Fig. 6. An example on the graph representation of reconfigurable battery packs.

To further include the output terminals into the graph representation, we extend the graph with the following three steps. First, two vertices n^+ and n^- are added to $\mathcal{V}$, representing the two output terminals of the battery pack. Then to capture the connectivity of the two terminals, we add edge $n^+ \rightarrow n_i$ and $n_i \rightarrow n^-(i = 1, 2, \ldots, N)$ to $\mathcal{E}$ if the output terminals can be directly connected to the ith cell. Note that for most existing battery pack designs, the output terminals can be directly connected to any cells through the backbone power buses. We assume this full connectivity of the output terminals in the rest of this article. At last, we extend the weight set $\mathcal{W}$ by setting $w(n^+) = w(n^-) = 0$.

Figure 6 illustrates an example of the extended graph with $N = 3$ based on the battery pack design in Alahmad et al. (2008). With this design, we can connect cell-0, cell-1, and cell-2 in series. As a result, directed edges $n_2 \rightarrow n^-$, $n_1 \rightarrow n_2$, $n_0 \rightarrow n_1$, and $n^+ \rightarrow n_0$ exist in the corresponding graph, as shown in Figure 6(b).

The graph captures the potential configurations of a given reconfigurable battery pack, and the out-degree of vertices is the metric to quantify the reconfigurability offered by the system. Note that any specific reconfigurable battery pack can be mapped to only one corresponding graph, while one graph may have multiple battery pack implementations. This is because the edges in the graph only reflect the logical connectivity among cells, but do not specify how to physically achieve such connectivity.

Also note that the graph model is a general abstraction of the battery pack, which could be tuned based on the specific problem under focus. For example, the edge direction should be defined based on whether a charging (e.g., He et al. (2016a)) or discharging (e.g., here in this article) problem is considered; the vertex weight could be defined as other cell properties such as their state-of-health as in He et al. (2015).

5 ADAPTIVE SYSTEM RECONFIGURATION

To effectively exploit the (likely limited) reconfigurability of battery packs, we first transform the problem of identifying the proper system configuration to a path selection problem in the corresponding battery graph. We then prove the NP-hardness of the transformed problem and show how it can be efficiently solved by a combination of DFS-with-pruning and 0-1 programming.

5.1 Problem Transformation

With the constructed graph, our problem can be transformed to identifying the maximal number of disjoint simple paths connecting n^+ and n^- with weight sum in $[V, (1 + \sigma)V]$, where σ is a control parameter that defines the closeness between the supplied and required voltages (i.e., the voltage difference should be no larger than σV). Specifically, we say a simple path connecting n^+ and n^- is

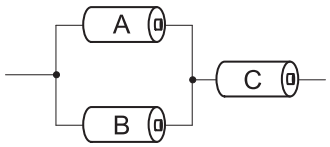


Fig. 7. Cell-C is shared by two strings: $A \rightarrow C$ and $B \rightarrow C$, leading to larger discharge current and faster SoH degradation.

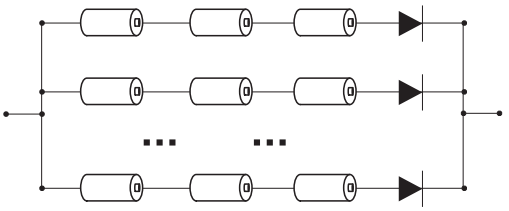


Fig. 8. Using diodes to regulate current direction and mitigate the voltage imbalance among strings.

feasible if the weight sum of its vertices is in $[V, (1+\sigma)V]$. The requirement on the disjoint paths is to avoid involving the same cell in multiple strings, which increases its discharge current (and thus reduces its capacity delivery) and degrades its state-of-health (SoH) faster (and thus makes the cell imbalance worse), as illustrated in Figure 7.

Note that the identified feasible strings may have different voltages, causing their imbalance when connecting in parallel. This may lead to the reverse charging of strings. We alleviate this potential issue with the control parameter σ, limiting the maximum imbalance of string voltages. Also, diodes are often used in practice to regulate the current direction and thus prevent the reverse charging, as illustrated in Figure 8.

5.2 Identifying the Configuration

We first show that identifying the maximum number of disjoint feasible paths is NP-hard, then based on two empirical observations, we propose a solution which is feasible in practice and achieves promising performance.

5.2.1 *NP-Hardness.*

Theorem 1. Given a directed graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathcal{W}\}$ and an interval $[V, (1+\sigma)V]$, finding the maximal number of disjoint simple paths in $\mathcal{G}$ with weight sum in $[V, (1+\sigma)V]$ is NP-hard.

Proof. The decision version of the above problem can be stated as follows: given a directed graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathcal{W}\}$, can we find $n (n = 0, 1, 2, \ldots)$ disjoint simple paths in $\mathcal{G}$ with weight sum in $[V, (1+\sigma)V]$? For the ease of description, denote this decision problem as Υ. We can prove that the *Longest Path Problem* in graph theory (i.e., finding a simple path of the maximum length in a given graph), a classic NP-complete problem, can be reduced to a special case of Υ by assuming (i) $n = 1$ and $\sigma = 0$; (ii) the weights of all vertices are 1; and (iii) V is an integer larger than 1. If a polynomial time algorithm Ψ exists for the special case of Υ, we can apply Ψ on the Longest Path Problem with path length increasing from 1 to $|\mathcal{V}|$, until no solution can be returned. This way, we solve the Longest Path Problem in polynomial time, contradicting with its NP-hardness. □

In the proposed solution, we first identify all feasible paths in the graph, then we find their largest disjoint subset. Each path in the returned subset represents a cell string, and all these strings are connected in parallel to support the load. Although the original problem is NP-hard, our solution is feasible in practice based on two empirical observations.

5.2.2 Step I: Finding All Feasible Paths. We identify the feasible paths in the graph via DFS-with-pruning. A computation time of $O(N^{N-1})$ is needed when using the basic DFS to identify all these paths.[2] However, the following two observations on battery systems assist in reducing the computation time in practice.

[2]Note the computation time to identify all the paths is different from graph traversal.

Observation 1. The set of supported configurations is limited for a given reconfigurable battery pack, which is captured by the battery graph.

This is because a higher reconfigurability is achieved at the cost of significantly increased system complexity. Denote the average out-degree of vertices as d; the computational complexity of DFS can be reduced to $O(Nd^{N-1})$, where the item N accounts for starting the search at each of the N vertices, and d^{N-1} accounts for the time to find all the paths starting at a given vertex, both of which can be further reduced with the following observation.

Observation 2. Because all the vertices weights are within range $[v_c, v_f]$, for a given load requirement, the number of vertices involved in a feasible path is limited.

Specifically, the minimum and maximum number of vertices in any feasible paths are $\lceil \frac{V}{v_f} \rceil$ and $\lceil \frac{(1+\sigma)V}{v_c} \rceil$, respectively. With this observation, the DFS only needs to examine a depth of at most $\lceil \frac{(1+\sigma)V}{v_c} \rceil$, and thus the worst-case complexity can be further reduced to $O(Nd^{\lceil \frac{V_{\max}}{v_c} \rceil - 1})$.

Besides reducing the worst-case complexity, the second observation also assists in pruning the DFS branches and thus reducing the average computation complexity from the following three aspects.

First, we do not need to apply the DFS at each of the N vertices. This is because an of important property of the graph's adjacent matrix—for any given graph $\mathcal{G}$, the element α_{ij}^k in the kth power of its adjacent matrix $\mathbb{A}^k$ indicates the number of paths from vertex i to j in $\mathcal{G}$ involving $(k+1)$ vertices. This way, we identify the possible starting points of feasible paths by finding the non-zero elements in $\mathbb{A}^k (k \in [\lceil \frac{V}{v_f} \rceil - 1, \lceil \frac{(1+\sigma)V}{v_c} \rceil - 1])$, at which should the DFS start only. It takes $O(kN^{2.37})$ to calculate $\mathbb{A}^k$ (Cormen et al. 2001).

Second, if the weights of the first i vertices in the current search are too small, it indicates that they cannot be part of any feasible paths. Specifically, denote $p(j)(j = 1, 2, \ldots, i)$ as the first i vertices included in the current search, then

$$\text{if} \quad V - \sum_{j=1}^{i} w(p(j)) > v_f \cdot \left(\left\lceil \frac{(1+\sigma)V}{v_c} \right\rceil - i \right) \quad \rightarrow \text{terminate the current search.}$$

Likewise, the current search can also be terminated if the weight of the first i vertices is too large

$$\text{if} \quad (1+\sigma)V - \sum_{j=1}^{i} w(p(j)) > v_c \cdot \left(\left\lceil \frac{V}{v_f} \right\rceil - i \right) \quad \rightarrow \text{terminate the current search.}$$

5.2.3 Step II: Finding the Largest Set of Disjoint Feasible Paths. Due to the requirement on disjoint paths, if we include a specific path into the system configuration, other paths with overlapping vertices should not be added to the configuration later. Our next step is to find the largest disjoint subset of all these feasible paths, assuming a total number of M feasible paths have been identified, which are denoted as $\mathcal{P} = \{path_1, path_2, \ldots, path_M\}$. We refer to the two paths as *conflicted* if they share at least one vertex, specifically, define a 0–1 matrix $A_{M \times N}$ as

$$a_{i,j} = \begin{cases} 1 & \text{if } n_j \in path_i \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

Then we define a *conflict matrix* $C_{M \times M}$ as

$$C_{M \times M} = \{conflict(i, j)\}, \tag{3}$$

and for any $path_i$ and $path_j$,

$$conflict(i,j) = 1 \Longleftrightarrow \exists k, a_{i,k} \cdot a_{j,k} = 1. \tag{4}$$

Thus our problem in finding the largest set of disjoint paths can be formulated as

$$\begin{aligned} &\max |\mathcal{P}^*| \;\; (\mathcal{P}^* \subseteq \mathcal{P}) \\ &s.t. \;\; \forall \;\; path_i, path_j \in \mathcal{P}^* \Rightarrow conflict(i,j) = 0, \end{aligned} \tag{5}$$

where $\mathcal{P}^*$ is the to-be-obtained largest path set. By further defining

$$x_i = \begin{cases} 1 & \text{if } path_i \in \mathcal{P}^* \\ 0 & \text{otherwise,} \end{cases} \tag{6}$$

we can transform the problem formulation in Equation (5) to

$$\max \sum_{i=1}^{M} x_i \;\; s.t. \;\; \forall j, \;\; \sum_{i=1}^{M} a_{i,j} \cdot x_i \leq 1. \tag{7}$$

The constraint in Equation (7) requires each vertex can be involved in at most one cell string to guarantee the disjoint paths. This is a classic 0-1 programming problem. As the DFS-with-pruning identifies all the feasible paths in the graph, we can see that the optimality (in terms of the size of the disjoint feasible path set) of the identified configuration only depends on how *optimal* a solution the 0-1 programming can obtain. Fortunately, efficient 0-1 programming solvers exist in the literature, and thus the near-optimality of the identified configuration can be guaranteed.

5.3 Switching to the Identified Configuration

After identifying the system configuration to support the load, the next step is to switch the system from its current state Ψ_c to thus-identified configuration Ψ^*. For the ease of description, we say an edge in the graph is *on* if the corresponding two cells are connected in series in Ψ_c, and *off* otherwise.

5.3.1 Safety and Efficiency Concerns when Switching Configurations. Although the paths in Ψ^* are disjoint, they are likely to share certain vertices with the paths in Ψ_c. As a result, cells may be shorted if directly applying Ψ^* onto Ψ_c. Figure 9 illustrates an example on this potential short of cells. Assume the path $n^+ \rightarrow n_2 \rightarrow n_4 \rightarrow n^-$ is adopted in the current configuration Ψ_c (Figure 9(a)) and the newly identified configuration Ψ^* is $n^+ \rightarrow n_2 \rightarrow n_1 \rightarrow n_4 \rightarrow n^-$ (Figure 9(b)); directly applying Ψ^* onto Ψ_c would lead to the combined configuration shown in Figure 9(c), in which n_1, n_2, and n_4 are shorted.

Another potential issue when applying Ψ^* onto Ψ_c is the degradation of energy efficiency. In the example shown in Figure 10, applying Ψ^* (Figure 10(b)) onto Ψ_c (Figure 10(a)) results in the configuration shown in the bottom of the figure, in which the paths $n^+ \rightarrow n_1 \rightarrow n_3 \rightarrow n^-$ and $n^+ \rightarrow n_2 \rightarrow n_4 \rightarrow n^-$ supply the loads in parallel. Note the second path does not exist in Ψ^* (e.g., because its weight sum is beyond $[V, (1+\sigma)V]$, increasing the regulator power loss).

A straightforward approach to eliminate these potential safety and efficiency issues is to turn all the *on* edges in Ψ_c into *off* state before applying Ψ^*. However, changing the edge states is completed by operating the corresponding switches, introducing additional energy consumption. Furthermore, a large number of electronic components operations also causes large latency in achieving Ψ^*.

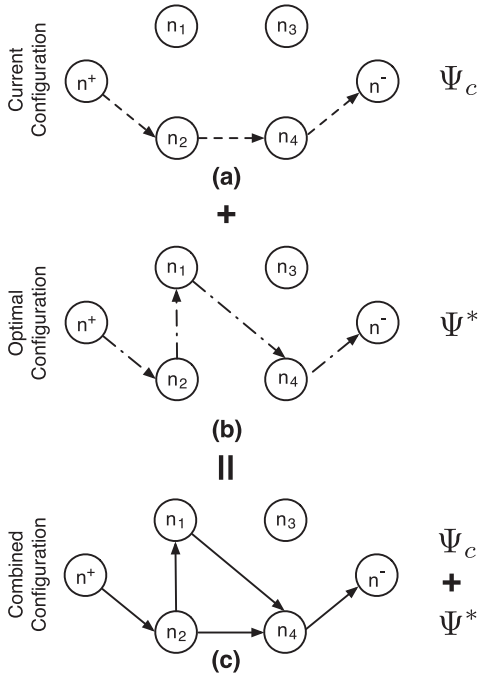


Fig. 9. Safety issue.

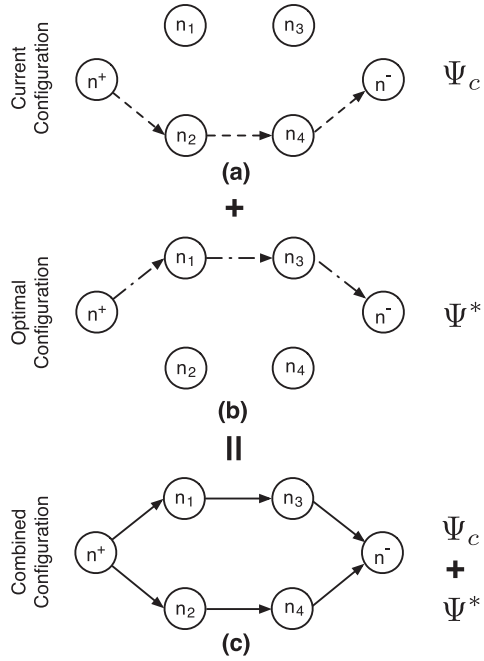


Fig. 10. Efficiency issue.

5.3.2 Two-Step Solution. In our design, we switch the system configuration from Ψ_c to Ψ^* by first eliminating the potential short of cells and then disconnecting the remaining strings in Ψ_c.

The following observation assists in identifying all the potential short of cells.

Observation 3. Removing the edge directions in $\mathcal{G}$ and denoting the resultant undirected graph as $\mathcal{G}'$, each circle in $\mathcal{G}'$ represents a potential short of cells in the battery pack.

With this fact, given a battery pack, we pre-identify all the circles in the corresponding $\mathcal{G}'$ with *Johnson's* algorithm (Johnson 1975),[3] and store these circles in the BMS. Then, every time to reconfigure the system, we combine Ψ^* with Ψ_c to check if all the edges of any pre-identified circles are in *on* state in the combined configuration. If not, no short circuit will happen when directly applying Ψ^* onto Ψ_c. Otherwise, we turn one *on* edge for each of these *all-on* circles into the *off* state (e.g., the edge $n_2 \rightarrow n_4$ in Figure 9). When determining which edge to turn *off*, we first check if any of these edges is included in a connected path in Ψ_c, and select these edges to turn *off* whenever possible. The edge is selected randomly otherwise.

Next, we need to disconnect all the paths that (i) $\in\Psi_c$, (ii) $\notin\Psi^*$, and (iii) are still connected after eliminating the potential short of cells. Specifically, for each of these still-connected paths, we select one of its edges that is not included in any paths of Ψ^*, and turn it to the *off* state. The feasibility of this approach is proved in the following theorem.

Theorem 2. For any connected paths in Ψ_c but not in Ψ^*, at least one of its edges is not included in Ψ^*.

[3]Johnson's algorithm identifies circles from a root vertex s in the subgraph induced by s and vertices larger than s in certain ordering of the vertices, with the help of a stack. Please refer to the original paper for algorithm details.

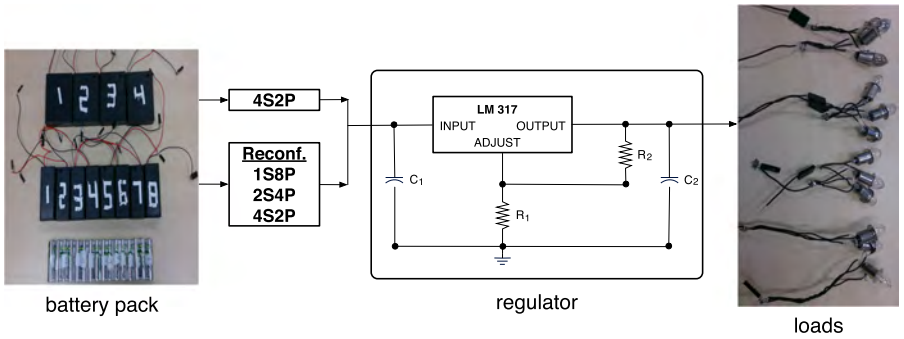


Fig. 11. Experiment methodology: forming the battery pack with 16 AA cells, regulating voltage with LM317 IC, and using 12 bulbs as the load.

Proof. We can prove the theorem by contradiction. For any connected path in Ψ_c but not in Ψ^*, if all its edges are included in Ψ^*, it means at least one pair of paths in Ψ^* share at least one common vertex, which contradicts with the condition that all paths in Ψ^* are disjoint. □

6 PERFORMANCE EVALUATIONS

We evaluate the adaptive reconfiguration via prototype-based experiments, large-scale simulations, and trace-driven emulations.

6.1 Prototype Experiments

6.1.1 Methodology. We build a prototype battery pack with 16 AA rechargeable cells with rated capacity of 2,450mAh, which are organized into eight modules each with two series connected cells. Similar to the reconfigurability offered by existing off-the-shelf products, the prototype supports a configuration set of {1S8P, 2S4P, 4S2P, 8S1P}.

We use 12 0.5A3.6W flashlight bulbs as the loads, which are organized into four 1S3P bulb modules. We randomly generate load traces in the form of $\{t_1, a_1\}, \{t_2, a_2\}, \ldots$, where t_i is the duration and a_i is the number of series connected bulb modules during t_i ($a_i \in \{1, 2, 3, 4\}$). Because the bulbs can operate under a wide voltage range (but with different lightness), for fair comparison, we impose an additional requirement that each bulb module requires a 2.5V input voltage, and the tolerable voltage gap between the supplied and required voltages is 1V unless otherwise specified. This way, the load required voltages vary from $2.5 \times 1 = 2.5$V to $2.5 \times 4 = 10$V. Note this also makes the configuration of 1S8P not feasible to support the load, due to insufficient voltage supply.

We adopt the 4S2P configuration as a baseline, which trades off between the highest voltage the system can supply (i.e., 4S) and the preference on small discharge current of cells (i.e., 2P). To match the supplied voltage with the load requirement, we implement an adjustable voltage regulator with the LM317 IC,[4] and connect it between the battery pack and the load to adjust the load input voltage to the required level. An overview of our experiment methodology is summarized in Figure 11.

We adopt the operation time, defined as the time from the start of the experiments to the time when the bulbs cannot be lightened anymore, as the metric to evaluate the system performance. The cells are initially fully charged with their corresponding chargers for each experiment.

6.1.2 Experiment Results. To investigate the performance of the adaptive configuration algorithm with different load conditions, we randomly generate three load traces with light, mild, and

[4]A simulator on the energy efficiency of LM317 is provided by TI (TI 2018).

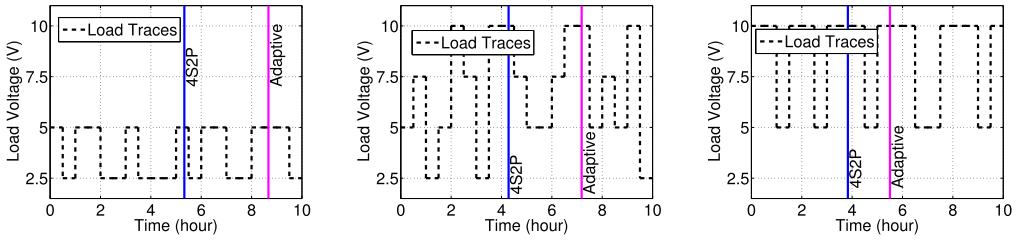


Fig. 12. Operation time with different loads.

heavy loads, respectively. Specifically, with light load, only one to two bulb modules are series connected as the load. The numbers of bulb modules adopted with the mild and heavy loads are randomly chosen from 1 to 4 and 2 to 4, respectively. In this way, the number of bulb modules used in the light, mild, and heavy loads are 1.5, 2.5, and 3 on average, respectively. The load lasting time t_i is set to 30min.

The operation times obtained with the two configuration methods for each load trace are shown in Figure 12. The advantage of the adaptive configuration over the baseline is obvious, and an average operation time increase of 3.06h is obtained over the three loads conditions. This operation time improvement is due to two reasons: first, by adaptively converting the supplied voltages to the load required levels, the energy loss on the voltage regulator is reduced; second, by minimizing the discharge current of individual batteries, more capacity can be delivered and the heat dissipation on other system components is also reduced.

Furthermore, we can observe that the lighter the loads, the more improvement can be obtained, which can be explained by the following two facts. First, with the 4S2P configuration, the lighter the loads, the larger the gap between the supplied and required voltages, degrading the regulator efficiency. Second, the lighter the loads, the fewer cells are needed to form a single series string to support the loads. This in turn offers more space for the adaptive configuration to identify more parallel strings, and thus reduces the battery discharge current.

Another observation from our experiment result is that with the 4S2P configuration, the temperature of the LM 317 IC easily rises to 44°C at the maximum. Such a high temperature not only indicates significant energy loss (and thus supports our design principle in matching supplied and required voltages), but also reduces the system stability.

6.2 Large-Scale Simulations

6.2.1 Trace-Based Battery Model. The analytical modeling of battery properties has been shown to be computationally expensive (Cloth et al. 2007; Rong and Pedram 2006; Rakhmatov and Vrudhula 2003; Moura et al. 2013). As a result, we use a trace-based method to track the cell states. We simulate a battery pack consisting of 2,900mAh Panasonic NCR18650 Li-ion cells, whose discharge curves with currents of 550mA, 2,750mA, and 5,500mA are provided in its data sheet (Figure 13(a)). The full and cutoff voltages are $v_f = 4.25$V and $v_c = 2.5$V, respectively.

To obtain more fine grained cell discharge traces, we fit the three traces with a 6° polynomial fitting (see Figure 13(b)), divide the current interval $[550, 5500]$mA into 99 intervals with a gap of 50mA each, and proportionally approximate the corresponding discharge traces. Note that we could further reduce the current gap to improve the approximation accuracy. A subset of the obtained discharge curves is shown in Figure 13(c) (not all the curves are shown for figure clarity).

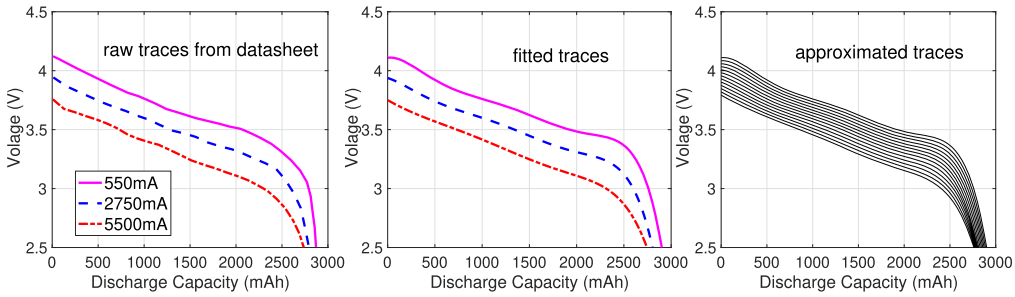


Fig. 13. Battery discharge traces adopted in the simulation: (a) raw traces from datasheet, (b) fitted traces, and (c) approximated traces.

6.2.2 Simulated Battery Packs. The simulated battery pack consists of 64 cells unless otherwise specified. The current drawn from each battery can directly reach two other cells on average (i.e., an average vertices out-degree of 2 in the graph). The initial cell voltages are randomly generated in the range of

$$[\alpha \cdot v_c,\ v_f], \tag{8}$$

where α is a control parameter that determines the cell voltage diversity and is set to 1.2 by default.

6.2.3 Load Traces. Similar to the load traces in the experiment, we randomly generate load traces in the form of $\{t_i, V_i, P_i\}$ for each load, where t_i is the lasting duration, and V_i and P_i are the required voltage and power of that trace, respectively. A unit time interval of 10min is adopted for loads' lasting time, i.e., t_i only takes the values of 10min, 20min, and so on. The system configuration is updated every 10min by first updating the cell voltages according to the traces in Figure 13, and then adaptively reconfiguring the battery pack. The required voltage V is randomly generated from 15 to 20V unless otherwise specified. The tolerable gap between the supplied and required voltage levels is 2.5V (i.e., v_c) by default. The required power P is randomly generated from 15V $\times$550mA = 8.25W to 20V $\times$5,500mA = 110W.

6.2.4 Baselines. We implement the following three system configurations as baselines.
Serial: The cells are series connected.
Parallel: $\sqrt{N}$ series strings with $\sqrt{N}$ cells each are formed.
Oracle: Given the range of all possible load required voltage, we can calculate how many cells are needed for a string to be able to support any potential load, and then we form as many as possible such strings. The remaining cells (those are not enough to construct another such string) are excluded from the final configuration. This oracle configuration maximizes the number of parallelly connected strings while guaranteeing each string is able to efficiently support the load requirement.

6.2.5 Performance Evaluation. We evaluate the impact of different system parameters on the performance of adaptive reconfiguration.

Cell Voltage Diversities. We first explore the impact of cell voltages on the operation time. The cells voltages are controlled by the parameter α in Equation (8), and a smaller α indicates both a higher voltage diversity among cells and a lower average cell voltage. The operation times with α varying from 1.1 to 1.7 are shown in Figure 14. Note that an α of 1.7 indicates the cell voltages are randomly generated from $[2.5 \times 1.7, 4.25] = \{4.25\}$, meaning all the cells are initially fully charged. The operation time increases as α increases because of higher initial cell voltages, and the adaptive reconfiguration outperforms the baselines in all the explored cases. Furthermore, we can see the

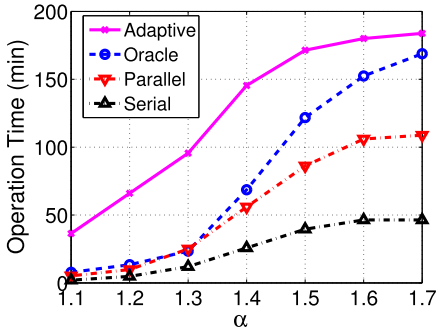


Fig. 14. Impact of initial cell voltages on system operation time.

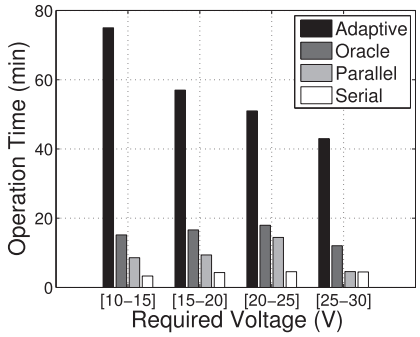


Fig. 15. Impact of required voltage on system operation time.

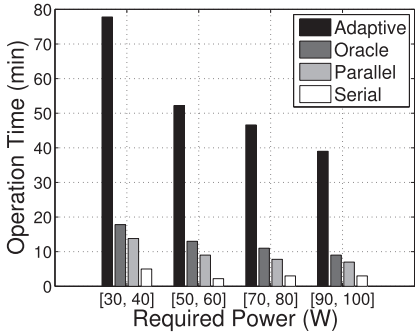


Fig. 16. Impact of required power on system operation time.

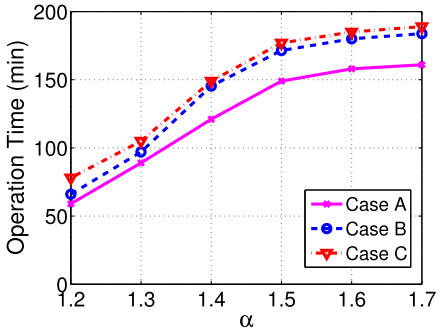


Fig. 17. Impact of reconfigurability distribution on operation time.

advantage of adaptive reconfiguration is more obvious with smaller α. This is because the smaller α is, the more likely that certain cells are close to depletion. Depleted cells significantly degrade the system performance if the configuration is not adjustable. On the other hand, the adaptive reconfiguration can bypass low-voltage cells when necessary, reducing their impact on system operation.

Load Requirements. The operation times with different load required voltages and powers are shown in Figure 15 and Figure 16, respectively. The load required power is set to 55W in Figure 15 and the required voltage is 20V in Figure 16. Again, significant improvement on the operation time can be observed with the adaptive reconfiguration, which is about 3×–5× of those obtained with the baselines. Furthermore, the operation time decreases as the loads become heavier. Also, Figure 15 shows that when the required voltage is relatively low (e.g., 10–25V), the operation time obtained with the non-reconfigurable baselines slightly increases with a higher voltage. This is because when the required voltage increases, the load current decreases with a fixed required power, which in turn leads to a longer operation time. However, as the required voltage further increases (e.g., in the range of 25–30V), the fixed configuration is more likely to fail to support the required voltage, and thus the operation time is reduced.

Reconfigurability. Next, we investigate the impact of system reconfigurability on the operation time. Fixing the average vertices out-degree as 3, we explore three cases on how the reconfigurability is distributed among cells, as shown in Table 1. The resulting operation times under these three cases are shown in Figure 17. The reconfigurability distribution significantly affects

Table 1. Three Cases of Configuration Flexibility Distributions

Vertices Out-degree	1	2	3	4	5
Case A	0.5	0	0	0	0.5
Case B	0.2	0.2	0.2	0.2	0.2
Case C	0	0	1	0	0

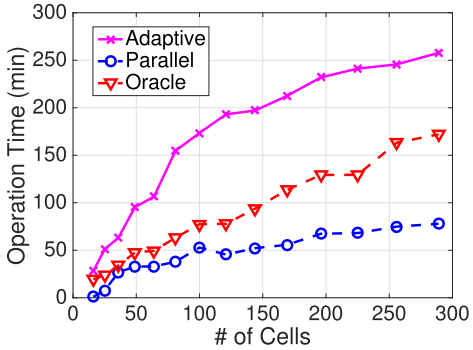


Fig. 18. Impact of N on system operation time.

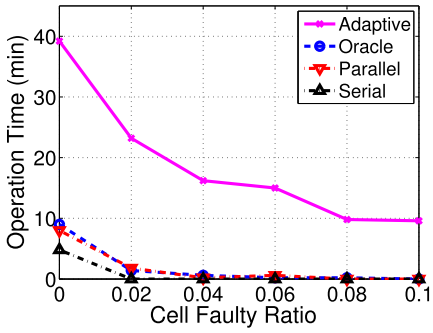


Fig. 19. Impact of cell failure on system operation.

system operation, which is longer when the reconfigurability is more evenly distributed among cells. This is because such an even distribution pattern alleviates the impact of bottleneck cells on the system performance. This observation serves as guidance in practical battery system design.

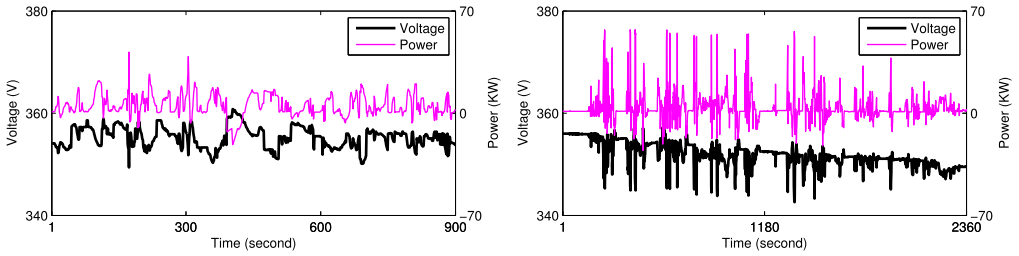
Number of Cells. The operation times obtained with different number of cells in the pack are shown in Figure 18. Intuitively, we observe that the operation time increases as more cells are included in the battery pack. The operation time decreases in the order of Adaptive, Oracle, and Parallel, because of the decrease in the number of parallel connected strings (and thus the increase of cell discharge current). Also, the advantage of reconfiguration pronounces with larger scales of battery packs. This is because more cells offer more choices to support the load with different configurations, which in turn enlarges the space for reconfiguration to take effect. Note that the Serial configuration is not plotted in Figure 18, in which case the total capacity delivery of batteries is always determined by the weakest cell.

Cell Failure Ratio. Battery cells fail over time, especially for large battery systems. The adaptive configuration can effectively mitigate cell failures as failed cells can be bypassed without significantly degrading the system performance. The operation time with cell faulty ratio from 0 to 10% of all the cells in the system are shown in Figure 19. Stronger robustness in system performance can be observed with the adaptive configuration, when compared with the three non-configurable baselines.

6.3 Emulations Based on EV Driving Trace

We further evaluate the adaptive reconfiguration via emulation based on empirically collected load traces when driving a Mitsubishi MiEV.

We collect two driving traces of about 900s and 2,400s each, containing the corresponding operation voltages and powers during that period as shown in Figure 20. We generate the load traces for our emulation with these two raw traces. First, both the discharge and charge of battery pack



(a) EV driving trace #1 (b) EV driving trace #2

Fig. 20. Driving traces with Mitsubishi MiEV.

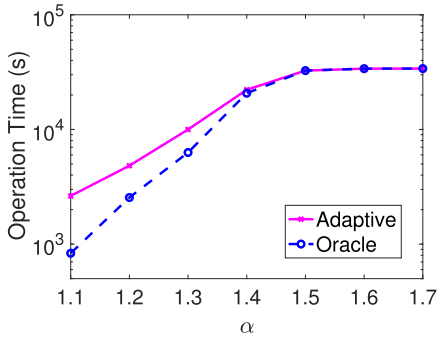


Fig. 21. Operation time with α.

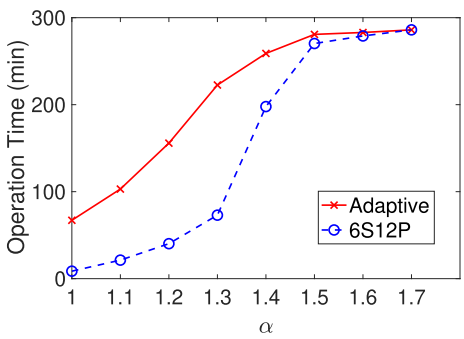


Fig. 22. Case-study on the AMP20 Module by A123.

happen during driving the EV. This is reflected in the traces that both positive (i.e., discharge of battery pack) and negative (i.e., charge of battery pack) values exist in the power traces. Because we only focus on the discharge scenario in this work, we simplify the evaluation by setting all the negative powers and the corresponding voltages in the traces to zero. This can be interpreted as no load is imposed on the battery pack during those time periods. Second, because the raw traces are relatively short in time, we combine them sequentially to form a 900 + 2,360 = 3,260s trace, and then repeat it for 10 times. As a result, a $\frac{3,260\times10}{60} \approx 550$min load trace is generated for our emulation.

The battery packs for electric vehicles normally take the hierarchical architecture: it can be divided into a set of battery modules which in turn consist of individual batteries. In our emulation, we form a battery pack consisting of 64 modules each with $16S4P$ connected cells. The battery discharge property conforms to the discharge traces shown in Figure 13.

Again, we take the non-reconfigurable Oracle baseline for comparison. The operation time with varying initial cell voltages is shown in Figure 21. Obvious advantage of the adaptive reconfiguration can be observed, especially when the variance of cell voltages is large. The two methods converge with α larger than 1.5. The operation time obtained with the two configuration methods converge as α increases, because in this case the battery pack has sufficient energy supply to survive the load even without reconfiguration.

6.4 Case Study on AMP20 Energy Module

Next we use a case study on the off-the-shelf 6S12P AMP20 Energy Modules from A123 to further explore the possibility of using reconfiguration to improve system operation. The AMP20 module supplies a voltage of 19.2V with $6 \times 12 = 72$ Li-ion cells (A123 2016). We randomly initiate these cells with various α and use them to simulate non-reconfigurable (i.e., cells are organized into

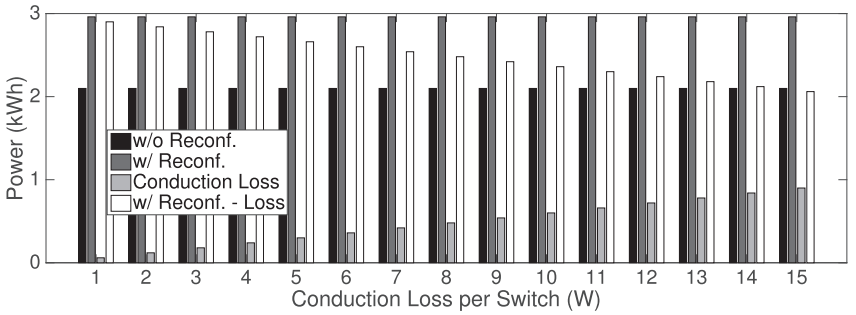


Fig. 23. Switches introduce additional conduction loss.

Table 2. Conduction Loss of NTD80N02 MOSFET

Current	On-Resistance	Conduction Loss
20A	7.5mΩ	3W
40A	7.5mΩ	12W
80A	5mΩ	32W

a fixed configuration of 6S12P) and reconfigurable battery packs. Figure 22 plots the resultant operation time: adaptive reconfiguration prolongs the operation time especially when the voltage heterogeneity among cells is high.

6.5 Power Loss on Switches

The reconfigurability of battery packs is achieved by equipping cells with switches, which introduces additional energy loss. Next, we explore the impact of the energy loss of switches on system performance. In the first case, a non-reconfigurable 6S12P A123 AMP20 Module is emulated. In the second case, a reconfigurable battery pack of the maximum reconfigurability (i.e., any given subset of cells can be organized into a string) is emulated with the same 72 cells. This maximum reconfigurability is to explore the upper bound of using reconfiguration to improve battery pack's power delivery. A switch is used to connect any two cells in series. The initial states of cells are randomly generated with $\alpha = 1.2$.

Figure 23 shows the deliverable power of the emulated battery packs with various conduction loss on switches, i.e., the dissipated power as current is conducted through the switches. As expected, reconfiguration improves the total power delivery of the battery pack, but part of which is lost due to heat dissipation on switches. This way, reconfiguration improves the battery pack's usable power when the conduction loss on switches is small, but may not be able to help when the conduction loss is large. As an example on the conduction loss of switches, Table 2 lists the 24V80A NTD80N02 MOSFET's on-resistance under different currents and the corresponding conduction loss. Referring to Figure 23, we can see that the reconfiguration improves the usable power when conducting currents of 20A and 40A, but may not help for the case of 80A current. Also note the conduction loss is only one type of overhead when equipping switches to cells; we will elaborate more on this in Section 7.

6.6 Evaluation Summary

As a summary, we make the following observations from these results. First, the proposed adaptive reconfiguration extends the system operation via reducing the energy loss and enhancing

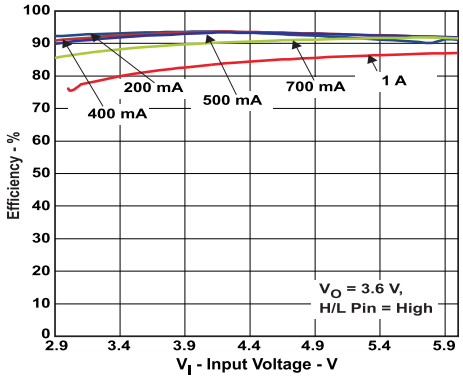


Fig. 24. Power efficiency of the switch-mode converter TPS62750 (TI 2016).

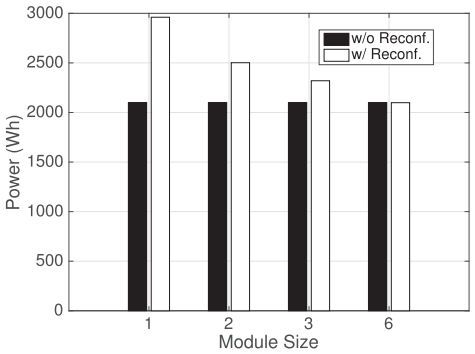


Fig. 25. Module-level reconfiguration: trading off between system complexity and reconfigurability.

the capacity delivery. Second, the advantage of the adaptive reconfiguration is pronounced when (1) the cell imbalance degree is high, (2) the load is light, and (3) the battery pack is large. Third, the adaptive reconfiguration demonstrates good tolerance of cell failures and thus improves system reliability. Last but not the least, higher reconfigurability has a diminishing effect in further extending the system operation, which is particularly important for identifying a good tradeoff between system's operation time and complexity.

7 PRACTICAL ISSUES

Here we discuss a few practical issues and future investigation directions.

7.1 Voltage Regulator or DC-DC Converter?

In this article, we assume voltage regulators are used in the system to ensure a stable voltage supply to the load, leading to a regulating efficiency that degrades with the gap between the supplied and required voltages proportionally. In practice, switch-mode DC-DC converter could be used to regulate battery packs' output voltage, which has relatively stable efficiency. However, voltage regulators do have a few advantages over DC-DC converters such as simplicity and noiseless, making them still widely used in real life. Also, the problem transformation in Section 5.1 is general enough to handle the cases when either regulators or DC-DC converters are used, by setting proper values of σ—smaller σ (and thus more constrained voltage ranges) when regulators are used, and larger σ (and thus slacker requirement on voltage) for DC-DC converters. This way, the proposed adaptive reconfiguration algorithm is applicable even when a DC-DC converter is used, and the system performance can still be improved via the principle on reducing cells' discharge current. Last but not the least, the efficiency of DC-DC converters, although relatively stable, is not perfect or totally independent with the input voltage from battery packs. Figure 24 plots the power efficiency of the switch-mode converter TPS62750 (TI 2016), showing pronounced dependency with input voltage as the current becomes larger. This dependency shows it is still feasible to reconfigure cells to supply an optimal voltage to the converter at which the highest efficiency is achieved (the improvement would be less significant when compared with the case with voltage regulators, though).

7.2 Module-Level Reconfiguration

This article is presented assuming the connectivity of individual cells can be actively adjusted. This cell-level reconfiguration raises the concern on scalability, especially when considering the various overheads of equipping switches. In practice, very-large-scale battery packs are normally of hierarchical structure. For example, the battery pack of Tesla S consisting of 16 modules, each of which consists of 6 bricks, and each brick consists of 74 cells. Module-level reconfiguration is a potential tradeoff between system complexity and reconfigurability. As an illustrative example, Figure 25 shows the power delivery of a 72-cell battery pack with module-level reconfiguration, where each module consists of x cells connected in series—smaller x means more modules are formed and thus more switches are needed. The case of a fixed configuration of 6S12P is also explored for comparison. Again, a voltage of 19.2V is required by the load. A clear tradeoff between reconfigurability (and thus power delivery) and system complexity is observed.

7.3 Overhead of Reconfiguration

The equipped switches in reconfigurable battery packs increase system overhead and cost, raising concerns on scalability. We have used a case study to explore the cost in terms of conduction loss in Section 6.5. Actually, the costs also come from other aspects such as increased system physical size/weight and switching delay, increased requirements on system thermal management due to heat dissipation, and so forth, especially for high-power systems such as EVs. The switching latency can be clearly observed in our prototype of a four-cell reconfigurable battery board based on the RTD34005 relays (please see He et al. (2016b)). Although various testbeds/prototypes of reconfigurable battery packs exist in the literature, these additional overheads/costs need to be thoroughly evaluated before the reconfigurable battery packs can be put into commercial use, leaving various open problems for further exploration.

7.4 Reconfiguration-Assisted Charging

This article focuses on exploiting the active reconfiguration of battery cells to improve their overall capacity delivery. Reconfiguration could be also explored to improve the charging of battery packs (e.g., our recent work (He et al. 2016a)) with a similar principle to this work (i.e., reconfiguring cells to achieve a desired voltage, which in turn allows the batteries to be (dis)charged with the desired current).

8 RELATED WORK

Investigations on exploring system reconfiguration to support dynamic load have been reported in Ci et al. (2012b) and Visairo and Kumar (2008), targeting on small battery systems such as mobile devices. In this work, we extend the investigation to large-scale battery systems.

Two necessary conditions must be satisfied to effectively and adaptively reconfigure the system. First, the system has to offer certain configuration flexibility on which the adaptive reconfiguration can operate. Research efforts have been devoted to effectively offer configuration flexibility with less additional costs (Alahmad et al. 2008; Kim et al. 2011b). For example, based on the design in Alahmad et al. (2008), four switches can connect a battery in series, parallel, or bypassed. Our work advances the state-of-the-art by proposing adaptive reconfiguration algorithms that return the desired system configuration based on the offered configuration flexibility and the real-time load requirements.

Second, the system has to be aware of individual battery conditions to perform the adaptive reconfiguration. Many works on battery modeling and simulation exist in the literature (Ceder et al. 2002; Chou et al. 2003; Cloth et al. 2007; Rong and Pedram 2006; Rakhmatov and Vrudhula

2003; Moura et al. 2013), with which the battery conditions could be estimated. However, most of these models are computationally extensive, and the simulators require practical parameters to implement. Furthermore, most of these models/simulators are for specific battery types, and thus their universalities are limited. Our proposed adaptive reconfiguration algorithms hide the complex low-level battery properties by focusing on two rules of thumb in identifying the desired system configurations: matching the supplied and required voltages and minimizing the battery discharge currents. A survey on reconfigurable battery packs can be found in Ci et al. (2016).

9 CONCLUSIONS

In this article, we have explored the adaptive reconfiguration of large-scale battery systems to dynamically provide the load's required voltages. Based on two empirically observed design principles, our approach hides the complex battery properties from engineering, and thus makes it more practical for implementation. Specifically, by abstracting the battery system into a graph representation, we have proposed an adaptive reconfiguration algorithm to identify the proper system configuration to support the load, which has been shown to significantly extend the system operation.

REFERENCES

A123. 2016. AMP20 Energy Modules. Retrieved on January 3, 2019 from http://www.a123systems.com/automotive/products/modules/.

Mahmoud Alahmad, Herb Hess, Mohammad Mojarradi, William West, and Jay Whitacre. 2008. Battery switch array system with application for JPL's rechargeable micro-scale batteries. *Journal of Power Sources* 177, 2 (2008), 566–578.

Battery. 2018. 18650 2800mAh Li-ion Battery. Retrieved on January 3, 2019 from https://www.amazon.com/Rechargeable-Mini-Butterball-Batteries-Flashlight/dp/B01GE3ZA10.

Gerbrand Ceder, Marc Doyle, Pankaj Arora, and Yuris Fuentes. 2002. Computational modeling and simulation for rechargeable batteries. *MRS Bulletin* (2002), 619–623.

P. Chou, C. Park, J. Park, K. Pham, and J. Liu. 2003. B#: A battery emulator and power profiling instrument. In *ISLPED'03*.

S. Ci, N. Lin, and D. Wu. 2016. Reconfigurable battery techniques and systems: A survey. *IEEE Access* 4 (2016), 1175–1189. DOI: https://doi.org/10.1109/ACCESS.2016.2545338

Song Ci, Jiucai Zhang, Hamid Sharif, and Mahmoud Alahmad. 2012a. Dynamic reconfigurable multi-cell battery: A novel approach to improve battery performance. In *APEC'12*.

Song Ci, Jiucai Zhang, Hamid Sharif, and Mahmoud Alahmadu. 2012b. A novel design of adaptive reconfigurable multiple battery for power-aware embedded networked sensing systems. In *GLOBECOM'12*.

L. Cloth, M. R. Jongerden, and B. R. Haverkort. 2007. Computing battery lifetime distributions. In *DSN'07*.

T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. 2001. *Introduction to Algorithms*. MIT Press.

Dimension Engineering. 2014. A beginner's guide to switching regulators. https://www.dimensionengineering.com/info/switching-regulators.

Liang He, Lipeng Gu, Linghe Kong, Yu Gu, Cong Liu, and Tian He. 2013. Exploring adaptive reconfiguration to optimize energy efficiency in large battery systems. In *IEEE RTSS'13*.

Liang He, Yu Gu, Cong Liu, Ting Zhu, and Kang G. Shin. 2015. SHARE: SoH-aware reconfiguration to enhance deliverable capacity of large-scale battery packs. In *ACM/IEEE ICCPS'15*.

Liang He, Linghe Kong, Siyu Lin, Shaodong Ying, Yu Gu, Tian He, and Cong Liu. 2016a. RAC: Reconfiguration-assisted charging in large-scale lithium-ion battery systems. *IEEE Transactions on Smart Grids* 7, 3 (2016).

Liang He, Shaodong Ying, and Yu Gu. 2016b. RTSS'13 Demo Video. https://www.youtube.com/watch?v=4oyj8nbGlk8.

D. B. Johnson. 1975. Finding all the elementary circuits of a directed graph. *SIAM Journal on Computing* 4, 1 (1975), 77–84.

Taesic Kim, Wei Qiao, and Liyan Qu. 2011b. Series-connected self-reconfigurable multicell battery. In *APEC'11*.

Taesic Kim, Wei Qiao, and Liyan Qu. 2012. A series-connected self-reconfigurable multicell battery capable of safe and effective charging/discharging and balancing operations. In *APEC'12*.

Younghyun Kim, Sangyoung Park, Yanzhi Wang, Qing Xie, Naehyuck Chang, Massimo Poncino, and Massoud Pedram. 2011a. Balanced reconfiguration of storage banks in a hybrid electrical energy storage system. In *ICCAD'11*.

Scot J. Moura, Jeffrey L. Stein, and Hosam K. Fathy. 2013. Battery-health conscious power management in plug-in hybrid electric vehicles via electrochemical modeling and stochastic control. *IEEE Transactions on Control Systems Technology* 21, 3 (2013), 679–694.

Neware. 2018. Neware Battery Testing System. Retrieved January 3, 2019 from http://newarebattery.com/.

Daler Rakhmatov and Sarma Vrudhula. 2003. Energy management for battery-powered embedded systems. *ACM Transactions on Embedded Computing Systems* 2 (2003), 277–324.

Delyan Raychev, Youhuizi Li, and Weisong Shi. 2011. The seventh cell of a six-cell battery. In *WEED'11*.

Peng Rong and M. Pedram. 2006. An analytical model for predicting the remaining battery capacity of lithium-ion batteries. *IEEE Transactions on Very Large Scale Integration Systems* 14, 5 (2006), 441–451.

TI. 2016. TPS62750: High efficiency step-down converter for USB applications. *Data Sheet.*

TI. 2018. LM317. Retrieved January 3, 2019 from http://www.ti.com/lit/ds/slvs044x/slvs044x.pdf.

K. Vatanparvar and M. A. Al Faruque. 2016. Eco-friendly automotive climate control and navigation system for electric vehicles. In *ICCPS'16.*

H. Visairo and P. Kumar. 2008. A reconfigurable battery pack for improving power conversion efficiency in portable devices. In *ICCDCS'08.*

S. Zeljkovic, T. Reiter, and D. Gerling. 2014. Efficiency optimized single-stage reconfigurable DC/DC converter for hybrid and electric vehicles. *IEEE Journal of Emerging and Selected Topics in Power Electronics,* 2, 3 (Sept. 2014), 496–506. DOI: https://doi.org/10.1109/JESTPE.2014.2305739

Received January 2018; revised July 2018; accepted October 2018