

Complete Proof for the NP-Completeness of the Vehicle-to-Vehicle Charging Model

Cláudio Gomes^{1,2,3}, João Paulo Fernandes⁴, Gabriel Falcao^{5,6},
Soumya Kar⁷, Sridhar Tayur⁸

¹Carnegie Mellon University, USA.

²FEUP, University of Porto, Portugal.

³LIACC, University of Porto, Portugal.

⁴New York University Abu Dhabi, United Arab Emirates.

⁵Instituto de Telecomunicações, Portugal.

⁶Department of Electrical and Computer Engineering, University of
Coimbra, Portugal.

⁷Department of Electrical and Computer Engineering, Carnegie Mellon
University, USA.

⁸Tepper School of Business, Carnegie Mellon University, USA.

Contributing authors: claudiogomes@cmu.edu; jpf9731@nyu.edu;
gabriel.falcao@uc.pt; soumyak@andrew.cmu.edu; stayur@cmu.edu;

1 Proof of NP-completeness

For the purposes of proving the NP-completeness of our model, we consider *V2V problem* as the *decision problem* in which an answer to a particular instance of the V2V model is always *True* if the constraints $Ax = b$, $l \leq x \leq u$ shown in equation (1) can be satisfied, and *False* otherwise.

$$(IP)_{A,b,l,u,f} : \begin{cases} \min f(x) \\ Ax = b, \quad l \leq x \leq u, \quad l, x, u \in \mathbb{Z}^n \\ A \in \mathbb{Z}^{m \times n}, \quad b \in \mathbb{Z}^m \end{cases}, \quad (1)$$

A proof for the NP-completeness of this model is presented by showing that the V2V problem belongs to the NP class, and by showing that the V2V problem is NP-hard. We prove the NP-hardness of the V2V problem by specifying and illustrating a polynomial reduction from the 3SAT problem to the V2V problem.

1.1 NP class

For any given decision problem, a *verifier* is an algorithm capable of verifying whether a solution τ given to an instance p of the problem is correct. Formally, a verifier is a function $V(p, \tau)$, which returns *True* when τ is correct and *False* when otherwise.

We define a *witness* x of a decision problem instance p as a solution to p such that $V(p, x) = \text{True}$, where V is a verifier of p . That is, a witness is a correct solution to a problem instance, which, once verified, let us conclude the problem instance is *True*. In order for the V2V problem to be in NP, we need to show that: there exists a polynomial-time verifier that enables us to verify that a witness attests to a particular instance of the problem to be *True*; and that the witness should be polynomial in the size of the problem instance.

For the V2V problem, we can define a polynomial-time verifier as the algorithm that checks whether the equality and inequality constraints $Ax = b$, $l \leq x \leq u$ shown in equation (1) are satisfied by a solution x . Concretely, this verifier of the V2V problem can be defined as

$$V_{\text{V2V}}((A, b, l, u), x) = \begin{cases} \text{True} & \text{if } Ax = b, l \leq x \leq u \text{ are satisfied,} \\ \text{False} & \text{otherwise,} \end{cases} \quad (2)$$

where (A, b, l, u) is a V2V instance and x is a solution to the instance. We remind the reader that this formulation uses the same notation as in equation (1) for simplicity. The matrix A and the vectors b, l , and u are represented as a series of constraints and the solution x is represented as the values for the binary decision variables X, Y , and Z , as well as the slack variables. V_{V2V} performs matrix multiplication and equality and inequality comparisons to check whether a given solution is correct for a given V2V instance. The matrix dimensions have been shown to be polynomial in terms of the number of vehicles, parking stations, meeting points, and timesteps. Moreover, the length of the vectors compared in the equality and inequality comparisons is the same as the number of columns in the matrix. Therefore, since these operations are known to be computable in polynomial time with respect to the dimensions of the matrix and of the vectors, V_{V2V} is a polynomial-time verifier.

Considering the verifier V_{V2V} , a witness x for it would be a set of values attributed to the binary decision variables X, Y , and Z , as well as the slack variables, that satisfy the constraints of the V2V instance (which is the set of matrices and vectors A, b, l , and u). In other words, a witness is a solution for a V2V instance that is able to route all the vehicles to their destinations while also meeting the V2V instance's path constraints, battery constraints, parking station charging constraints, and V2V charging constraints.

Any solution of a V2V instance is a set of values attributed to its binary decision variables X, Y , and Z , as well as the slack variables. By definition, the length of these variables is the same as the number of columns of the matrix A of the instance. Again,

by definition, a witness is a solution of a V2V instance. Therefore, a witness of a V2V instance is polynomial in terms of the size of the problem instance (A , b , l , and u).

We have shown that there exists a verifier that enables us to verify that a witness attests to a particular instance of the problem to be *True* in time polynomial in the size of the problem instance; and that the size of the witnesses is polynomial in the size of the problem instance. Thus, we have proven that the V2V problem is in NP.

1.2 Reduction from the 3SAT problem

A problem is considered NP-hard if we can reduce all the problems in NP to it in polynomial time. To prove that the V2V problem is NP-hard, it suffices to show a polynomial-time reduction to it from a known NP-hard problem. More concretely, we will show that the V2V problem is NP-hard by performing a polynomial-time Karp reduction from the 3SAT problem.

1.2.1 3SAT Problem

The 3SAT problem is a decision problem that is known to be NP-hard and defined as follows:

Definition 1. *The 3SAT problem accepts as input a Boolean formula in **conjunctive normal form** such that there are at most three literals in each clause. There may be any number of atoms and clauses. The output is **True** if the formula is **satisfiable** and **False** otherwise.*

We remind the reader that a literal is one of two types in the context of the 3SAT: *positive literal* when it is just the atom; and *negative literal* when it is the negation of the atom. Moreover, a Boolean formula in the conjunctive normal form is a conjunction of one or more clauses, where each clause is a disjunction of literals. Finally, a Boolean formula is considered satisfiable when there is an assignment of values to the atoms that will satisfy it. A Boolean formula is considered to be satisfied by an assignment of values to its atoms when it is *True* under that assignment. We illustrate an instance of the 3SAT next:

$$(x_1) \wedge (x_2 \vee x_3) \wedge (\neg x_2 \vee \neg x_3) \wedge (x_1 \vee \neg x_2 \vee x_3) \quad (3)$$

Going back to the context of our NP-completeness proof of the V2V problem, we define a verifier for the 3SAT problem as follows:

$$V_{3SAT}(p, x) = \begin{cases} True & \text{if } p \text{ is } True \text{ under } x, \\ False & \text{otherwise,} \end{cases} \quad (4)$$

where p is the 3SAT instance, defined by its Boolean formula, and x is a solution to the instance, defined as a list of (atom = truth value) pairs, with one pair for each atom. Since the 3SAT problem is in NP, this verifier can be executed in a time polynomial to the number of atoms and clauses (which is easy to see how, given it only needs to compute a polynomial number of bitwise operations). Now that a polynomial-time verifier V_{3SAT} has been defined, we can define a witness x of a 3SAT instance p as a list of (atom, truth value) pairs such that $V_{3SAT}(p, x) = True$. In other words, the witness x of a 3SAT instance is an assignment of values to the atoms such that it satisfies the

Boolean formula of the instance. As an example, $(x_1 = \text{True}, x_2 = \text{False}, x_3 = \text{True})$ is a witness for the 3SAT instance shown in (3).

1.2.2 Reduction Steps

We show next the reduction steps from the 3SAT problem to the V2V problem. Let us suppose that we are given an instance of 3SAT with n atoms x_1, \dots, x_n and m clauses c_1, \dots, c_m . Then we construct the space network graph $G = (N, A)$ and the other parameters for the equivalent V2V instance as follows:

- For each clause c_j , we create a node $sat_j \in M$, which is a meeting point ($M \subset N$), and a node $f_j \in N$, which are connected by a direct edge from sat_j to f_j .
- For each atom x_i , we create a starting node $s_i \in N$. Next, we create a node $true_i \in M$ and a node $false_i \in M$, which are meeting points. Afterwards, we create directed edges from s_i to $true_i$ and to $false_i$.
- For each clause c_j and for each atom x_i , if c_j has a literal of x_i , we create a directed edge from $true_i$ to f_j and from $false_i$ to f_j ; if c_j has a positive literal of x_i , we create a directed edge from $true_i$ to sat_j ; and if c_j has a negative literal of x_i , we create a directed edge from $false_i$ to sat_j .
- For each atom x_i , we denote as k_i the number of literals it has among all clauses. Next, we create k_i EVs $v_{i,1}$ to v_{i,k_i} , all located on s_i . For each vehicle $v_{i,o}$, $o \in \{1, \dots, k_i\}$, we set its final destination as f_j , where c_j is the clause that contains the o^{th} literal of the atom x_i . $\text{SOC}_{v_{i,o}}$ is 1 if $o \neq 1$ and $3k_i + 1$ if otherwise.
- For each clause c_j , we create an EV v_j^{sat} located at sat_j and whose final destination is f_j . $\text{SOC}_{v_j^{sat}}$ is 0.
- For each EV v , we set $e_i = 1$ (energy that EV v provides to another EV in a timestep) and MAXSOC_v as a sufficiently large number, such as $3m + 1$.
- All the arcs (or directed edges) created have a time duration of 1 and a traversal cost of 1.
- Finally, we define $T = 3K + 6$, where K is the greatest number of literals any atom x_i has among all clauses ($K = \max(k_1, \dots, k_n)$).

With the construction steps listed above, we have defined all the parameters required for the input of a V2V problem. Therefore, the graph G we just constructed will be transformed into a time-space network graph G_{TS} , finalizing the setup of a V2V instance from a 3SAT instance.

1.2.3 Proof of Correctness for the Reduction

In order to show that our reduction from the 3SAT problem to the V2V problem is correct, we prove that the constructions steps shown are equivalent to a function f mapping instances of the 3SAT problem to instances of the V2V problem, such that:

$$\text{answer to } p \text{ is } \text{True} \iff \text{answer to } f(p) \text{ is } \text{True}, \quad (5)$$

where p is an instance of the 3SAT problem. f is also known as a *Karp reduction* from the 3SAT problem to the V2V problem.

Since we have proven that both the 3SAT problem and the V2V problem are in NP, we can leverage the definition of NP to show that a mapping function f is a Karp reduction from the 3SAT problem to the V2V problem if and only if for all instances p of the 3SAT problem the following is true:

$$\begin{array}{ll} \text{There exists a polynomial-} & \text{There exists a polynomial-} \\ \text{size witness } x \text{ such that } & \text{size witness } z \text{ such that} \\ V_{3\text{SAT}}(p, x) = \text{True.} & V_{\text{V2V}}(f(p), z) = \text{True.} \end{array} \quad (6)$$

where $V_{3\text{SAT}}$ and V_{V2V} are the previously defined polynomial-time verifiers for the 3SAT problem and the V2V problem, respectively.

We present a series of lemmas and their proofs next. These lemmas culminate into a theorem that shows that our reduction is a Karp reduction and, consequently, that the V2V problem is NP-hard (and thus NP-complete).

Lemma 1. *The reduction from the 3SAT problem to the V2V problem shown in 1.2.2 is polynomial in time and size in terms of the number of atoms and clauses.*

Proof. Let us suppose that we are given an instance of 3SAT with n atoms and m clauses. As shown by the construction steps, the reduction creates an instance of the V2V problem that uses a space network graph with $3n + 2m$ nodes and, at most, $2n + 10m$ arcs. As an example to clarify the number of arcs, a clause c_j implies having an arc from sat_j to f_j and may have three atoms x_i, x_j, x_k with positive literals in it and, thus, also have arcs from the nodes $true_i, true_j, true_k, false_i, false_j, false_k$ to f_j , as well as arcs from the nodes $true_i, true_j, true_k$ to sat_j . \square

Lemma 2. *If the answer to a 3SAT instance is **True**, then the answer to its reduction to a V2V instance is also **True**.*

Proof. Consider a 3SAT instance p with n atoms and m clauses. Let us assume that the answer to p is *True*. Therefore, there exists a polynomial-size witness x such that $V_{3\text{SAT}}(p, x) = \text{True}$. We remind the reader that a witness for a 3SAT instance is an assignment of values to the atoms of the Boolean formula of the instance, in the form of a list of (atom = truth value) pairs. Hence, for each variable x_i of the instance, we have the pair (x_i, True) or the pair (x_i, False) in the list. This witness x can be transformed into a witness z such that we have $V_{\text{V2V}}(f(p), z) = \text{True}$:

- For each atom x_i , if its pair is *True*, we move all EVs $v_{i,1}, \dots, v_{i,k_i}$ from s_i to $true_i$, otherwise to $false_i$. This step happens from $t = 0$ to $t = 1$ and takes one unit of energy from each vehicle involved.
- For each atom x_i and $o \in \{2, 3, \dots, k_i\}$, vehicle $v_{i,1}$ will charge $v_{i,o}$ with 3 units of energy from $t = 3o - 5$ to $t = 3o - 2$.
- For each atom x_i and $o \in \{1, 2, \dots, k_i\}$, if there is an arc between the current node of $v_{i,o}$ and sat_j , where c_j is the clause that contains the o^{th} literal of the atom x_i , then $v_{i,o}$ will move to sat_j , charge v_j^{sat} with one unit of energy, and then move to f_j ; otherwise, if there is no arc between the current node of $v_{i,o}$ and sat_j , then $v_{i,o}$ will be in the same node for two timesteps and then move to f_j . If $o \geq 2$, $v_{i,o}$ will perform these actions from $t = 3o - 2$ to $t = 3o + 1$. $v_{i,1}$ will perform these actions

last, from $t = 3k_i + 1$ to $t = 3k_i + 4$. These actions consume 1 or 3 units of energy from each vehicle involved, depending on whether they go directly to f_j .

- After being charged by all possible vehicles, all $v_1^{\text{sat}}, \dots, v_m^{\text{sat}}$ will move to f_1, \dots, f_m , respectively. This happens during the last timestep, from $t = T - 2$ to $t = T - 1$ (from $t = 3K + 4$ to $t = 3K + 5$, $K = \max(k_1, \dots, k_n)$).

In order for $V_{\text{V2V}}(f(p), z) = \text{True}$, witness z will have to assign the required actions in the V2V instance $f(p)$ such that all the EVs reach their destinations. We prove that this transformation from the witness x is, indeed, a valid witness z :

- Since $T = 3K + 6$ and $k_i \leq K, \forall i \in \{1, \dots, n\}$, all the steps above do not exceed the number of timesteps specified for the V2V instance.
- For each atom x_i , if its pair is *True*, $v_{i,1}$ arrives to true_i with $3k_i$ units of energy, all the vehicles $v_{i,2}, \dots, v_{i,k_i}$ arrive to true_i with 0 units of energy. After arrival, they each get 3 units of energy from $v_{i,1}$, which loses $3(k_i - 1)$ units of energy. All in all, all the vehicles $v_{i,1}, \dots, v_{i,k_i}$ leave true_i with 3 units of energy. By the same logic, for each atom x_i , if its pair is *False*, all the vehicles $v_{i,1}, \dots, v_{i,k_i}$ leave false_i with 3 units of energy. Hence, for each atom x_i , each of the vehicles $v_{i,1}, \dots, v_{i,k_i}$ has enough of units of energy in case it moves directly to its respective f_j , or even if it moves to its respective sat_j , charges its respective v_j^{sat} , and then moves to its destination f_j .
- It remains to prove that, for each clause c_j , v_j^{sat} reaches their destination f_j , which is equivalent to proving that v_j^{sat} receives at least one unit of energy, as it only needs one unit to move to f_j . We have that all the clauses of the 3SAT instance p are satisfied by the list of (atom = truth values) pairs defined by the witness x . Therefore, there is at least one literal in each clause c_j that is satisfied by some atom x_i with an assigned truth value. Moreover, considering the construction steps for the instance $f(p)$, there will be an arc between true_i and sat_j if the literal is positive, and an arc between false_i and sat_j if otherwise. We also have that, by mapping, if o is the index of the literal for the atom x_i , the vehicle $v_{i,o}$ is moved from s_i to true_i if the value assigned to x_i is *True*, and to false_i if otherwise. Hence, $v_{i,o}$ will move into a node that coincides with the arc that connects that node to sat_j . Consequently, $v_{i,o}$ will move to sat_j and charge its respective v_j^{sat} before moving to its respective f_j . All in all, it is proven that, for each v_j^{sat} , there is at least one vehicle charging one unit of energy to it and, thus, v_j^{sat} is able to reach f_j .

□

Lemma 3. *If the answer to a V2V instance $f(p)$ reduced from a 3SAT instance is **True**, then all its witnesses, for any atom x_i , will have all the vehicles $v_{i,1}$ to v_{i,k_i} pass through one and only one of the meeting points true_i or false_i .*

Proof. We remind the reader that, for any atom x_i of the 3SAT instance, the reduction to a V2V instance will have k_i related vehicles $v_{i,1}$ to v_{i,k_i} , where k_i is the number of literals the atom x_i has among all clauses. In order for each of these vehicles to reach their destination f_j , they need to traverse at least two arcs, as the shortest path goes from the starting node s_i to one of the meeting points true_i or false_i , and then from that meeting point to f_j . Therefore, each vehicle needs at least two units of energy to

reach their destination. By construction, $v_{i,1}$ has an associated $\text{SOC}_{v_{i,1}} = 3k + 1$, while $v_{i,2}$ to v_{i,k_i} have an associated $\text{SOC}_{v_{i,2}} = \dots = \text{SOC}_{v_{i,k_i}} = 1$. Hence, all vehicles but one need to be charged one unit of energy in order to reach their destination. For this to be possible, they need to traverse to one of the meeting points true_i or false_i . When all k_i vehicles move to a meeting point true_i or false_i , they consume one unit of energy, all arriving to the meeting point with 0 units available, except for $v_{i,1}$, which has $3 \times k_i$ units available. Hence, in order for all vehicles to reach their destination, at least k_i of the $3k_i$ units stored in $v_{i,1}$ need to be shared among all the vehicles. This implies that vehicles need to be in the same meeting point true_i or false_i as $v_{i,1}$ in order to get a unit of energy, since vehicles in true_i cannot move to false_i and vice-versa (there is no path between true_i and false_i , as the only way to enter into these nodes is via the arcs from s_i and there is no arc to enter into s_i). \square

Lemma 4. *If the answer to a V2V instance reduced from a 3SAT instance is **True**, then the answer to that 3SAT instance must be **True**.*

Proof. Let us assume that we have a witness z for the V2V instance $f(p)$, such that $V_{\text{V2V}}(f(p), z) = \text{True}$:

- Then, by Lemma 3, for each atom x_i of the instance p , we have that all EVs $v_{i,1}$ to v_{i,k_i} go from s_i to the same meeting point true_i or false_i .
- Moreover, by definition, for each clause c_j of the instance p , we have that the vehicle v_j^{sat} arrives to its destination f_j .

The second item implies that, for each node sat_j , we have at least one arc from some node true_i or false_i to sat_j , and that at least one vehicle $v_{i,o}$ moved from that node to sat_j in order to give one unit of energy to v_j^{sat} , where o is the index of the literal for the atom x_i of the instance p , enabling v_j^{sat} to reach its destination.

Together with the first item, we have that the meeting points true_i and false_i chosen by a witness z for the vehicles corresponding to each atom x_i can be transformed into a witness x of the 3SAT instance p , by assigning a value of *True* to x_i when its vehicles $v_{i,1}, \dots, v_{i,k_i}$ pass through true_i , and a value of *False* when otherwise. This transformation is a witness of p , since, by construction, the existence of an arc from some node true_i or false_i to the node sat_j implies that a positive or negative literal of some atom x_i exists in the clause c_j , respectively, and that the same clause c_j is satisfied by the atom x_i , as it has the same value as that literal. Hence, the existence of a witness for p implies that the answer to p is also *True*, since we have $V_{\text{3SAT}}(p, x) = \text{True}$. Thus, the answer to the 3SAT instance p cannot be *False* when the answer to its reduction to a V2V instance is *True*. \square

Theorem 5. *The answer to a 3SAT instance p is **True** if and only if the answer to its reduction to a V2V instance $f(p)$ is **True**. The reduction f is a Karp reduction.*

Proof. The proof follows from combining Lemmas 1, 2 and 4. Thus, by definition, the reduction f is a Karp reduction. \square

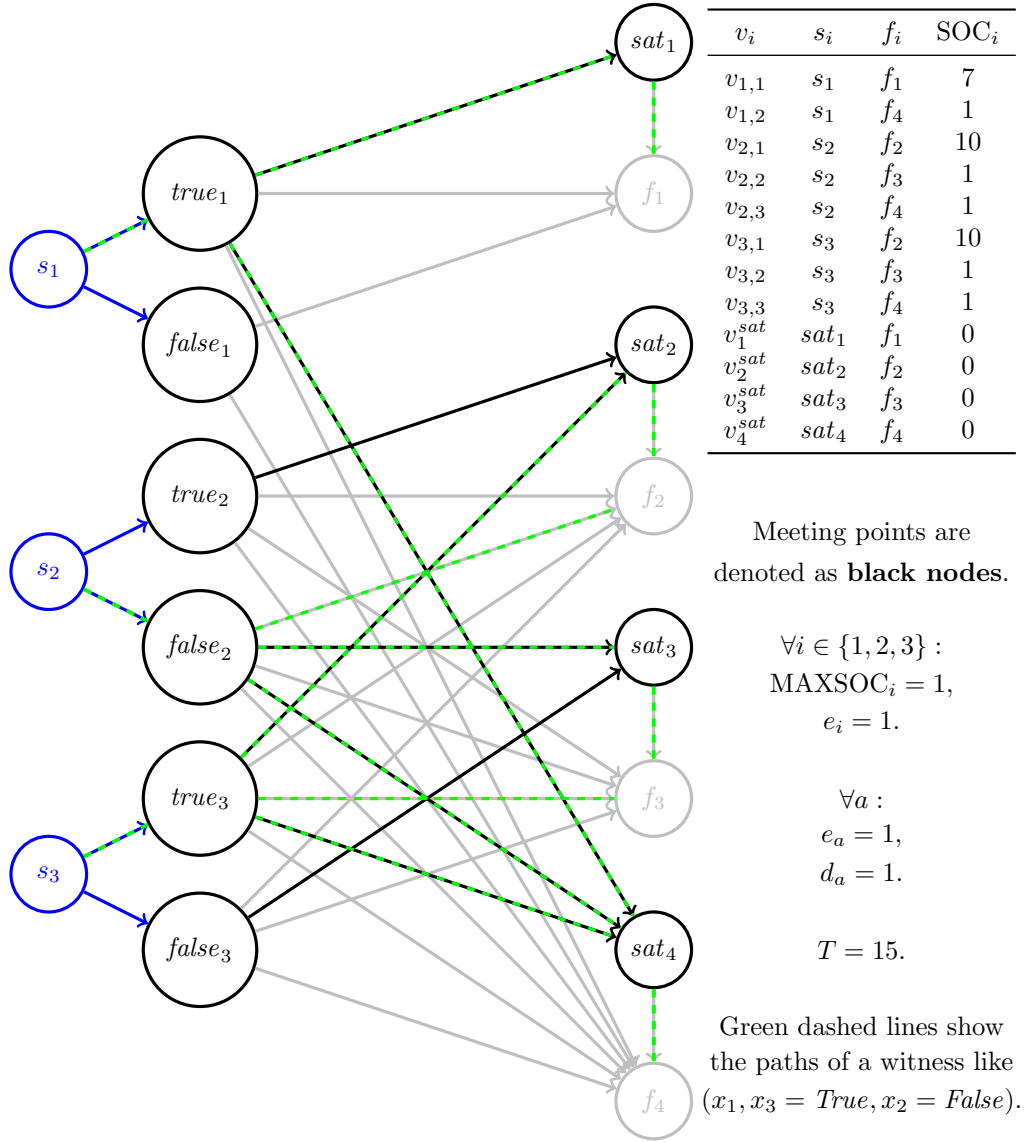


Fig. 1 Illustration of a reduction from a 3SAT instance to a V2V instance.

1.3 Proof of NP-completeness

Now that we have proved that the V2V problem is in NP and is NP-hard, then, by definition, the V2V problem is NP-complete. As an aid for the reader to understand the proof, Figure 1 shows an illustration of the reduction from the 3SAT instance shown in equation (3).

2 Acknowledgements

We express our special appreciation to Anthony Karahalios and Oleg Plisov for their proof-checking and generous feedback.