AUSTRALASIAN JOURNAL OF COMBINATORICS
Volume **62(1)** (2015), Pages 45–58

A linear-size conversion of HCP to 3HCP

V. Ejov M. Haythorpe* S. Rossomakhine

School of Computer Science, Engineering and Mathematics
Flinders University
Sturt Road, Bedford Park, SA 5042
Australia

vladimir.ejov@flinders.edu.au serguei.rossomakhine@flinders.edu.au
michael.haythorpe@flinders.edu.au

Abstract

We provide an algorithm that converts any instance of the Hamiltonian cycle problem (HCP) into a cubic instance of HCP (3HCP), and prove that the input size of the new instance is only a linear function of that of the original instance. This result is reminiscent of the famous SAT to 3SAT conversion by Karp in 1972. Known conversions from directed HCP to undirected HCP, and sub-cubic HCP to cubic HCP are given. We introduce a new subgraph called a 4-gate and provide a procedure that converts any sub-quartic instance of HCP to a sub-cubic instance. Finally, we describe a procedure to convert any graph to a sub-quartic graph, and use the previous results to provide an algorithm which converts HCP to 3HCP with only linear growth in the instance size.

1 Introduction

The Hamiltonian cycle problem (HCP) is a famous NP-complete graph theoretic decision problem that can be described simply: given a graph Γ, determine whether it contains any simple cycles containing all vertices in the graph, or not.

Throughout this paper, when referring to HCP in its traditional form, we will use the expression *general HCP*. An *instance* of general HCP takes the form of a (possibly directed) simple graph, which is defined by its vertices and (directed) edges. We refer to the sum of the number of vertices and the number of edges of an instance as the *input size* of the instance. So if a graph contains n vertices and m edges, the input size is $n + m$. However, any meaningful instance of HCP will have at least as many edges as vertices, or else it is trivially non-Hamiltonian. For this reason, when talking about the order of the input size, it suffices to merely consider the number of edges in the graph.

* Corresponding author.

Although the definition of general HCP permits any simple graph as an instance, a natural idea is to adopt restricted definitions of HCP in which only graphs that satisfy certain properties are to be considered. For some such restrictions, HCP is known to remain NP-complete. The first of these was proved by Karp [16], who showed that any general HCP instance can be converted to an equivalent instance which only contains undirected edges. The input size of the new instance is a linear function of that of the original instance. We will use the expression *undirected HCP* to describe the restricted version of HCP where only undirected instances are permitted. For the sake of completeness, we include the reduction here.

Suppose we have a directed graph containing n vertices, which forms an instance of general HCP. We can produce a new graph containing $3n$ vertices, and add edges to it using the following algorithm.

General HCP to Undirected HCP Conversion Procedure

1. Add edges $(3i-1, 3i-2)$ and $(3i-1, 3i)$ for all $i = 1, \dots, n$.
2. For each (directed) edge (i, j) in the original graph, add edge $(3i, 3j-2)$.

In the above procedure, a new graph instance is constructed from scratch. For convenience, however, for the remainder of this manuscript we will think of such procedures as having replaced certain components of a graph with new components, whose constructions depend upon the components they are replacing, as displayed in Figure 1.1. If the original instance has maximum in-degree r and maximum out-degree s, the new undirected instance will have maximum degree of $\max(r, s) + 1$. It is easy to see that there is a 1-1 correspondence between Hamiltonian cycles in the original instance and Hamiltonian cycles in the equivalent undirected graph.

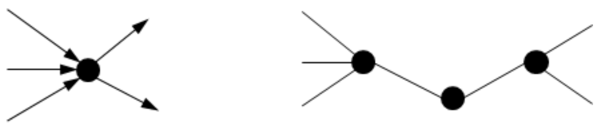

Figure 1.1: A vertex with adjacent directed edges, and the corresponding undirected subgraph which replaces it.

Throughout this manuscript, we will refer to conversions as being *polynomially-growing*, depending on the degree of the polynomial that describes the new input size as a function of the old input size. For example, the above procedure describes a *linearly-growing conversion*, because the resultant graph has input size which is a linear function of the original input size. Specifically, in the above conversion, if the original graph had n vertices and m (directed) edges, the new undirected graph will contain $3n$ vertices, and $m + 2n$ undirected edges.

In addition to the conversion from general HCP to undirected HCP, it was proved by Garey et al. [10] that even if HCP is restricted to only instances that are undirected, cubic, planar and 3-connected, the problem is still NP-complete. Of course, this implies that one may restrict HCP instances to any subset of those four conditions and the problem will remain NP-complete. In this manuscript we will be primarily interested in cubic, undirected HCP instances. We will refer to HCP restricted to such instances as *cubic HCP* (note that the undirected condition is implied in this title). Cubic HCP is a widely studied problem in its own right, with Barnette's conjecture [2] that every bipartite, planar, 3-connected cubic graph is Hamiltonian still open. More recently, Eppstein [7] conjectured that undirected cubic graphs with n vertices have at most $2^{n/3}$ Hamiltonian cycles, with Gebauer [11] providing the best proven bound to date of approximately 1.276^n. Eppstein [7] also provided an algorithm (of exponential time complexity) for finding Hamiltonian cycles in cubic graphs. Another open conjecture by Filar et al. [9] is that almost all non-Hamiltonian cubic graphs are bridge graphs, and may therefore be easily detected. Royle [19] maintains an online database of exhaustive sets of small cubic graphs containing various properties.

In practice, however, the result of Garey et al. [10] is inefficient, as the reduction in [10] is not from general HCP, but rather from boolean satisfiability (SAT) in conjunctive normal form with clauses of size 3 (3SAT). Currently, the best known conversion from general HCP to SAT is a cubically-growing conversion [18], and the conversion of SAT to 3SAT is a linearly-growing conversion [16]. Converting from 3SAT to cubic, planar, 3-connected HCP using the method by Garey et al. requires a quadratically-growing conversion; however, if we drop the requirement of planarity, the conversion reduces to linearly-growing. So, to convert a general HCP instance into a cubic HCP instance via the approach in [10], we must first convert to SAT, then to 3SAT, and finally to cubic HCP, which results in a cubically-growing conversion.

Although it is generally accepted that conversions which result in polynomial growth are, in some sense, efficient, very little attention seems to have been paid to the degree of the polynomial which describes the growth. From a practical perspective, it seems unlikely that any one problem framework could be sufficiently simpler to solve difficult instances in than another to justify using conversions that induce anything larger than linear growth. From a theoretical perspective, it is an interesting, and largely unexplored[1] line of research to see which NP-complete problems may be converted to one another via linearly-growing conversions. To that end, we now describe a new approach to convert directly (that is, we remain within the scope of HCP for the entirety of the conversion) from general HCP to cubic HCP, using a linearly-growing conversion. We begin by revising the well-known conversion for sub-cubic (undirected) HCP to cubic HCP. We then provide a new conversion for sub-quartic HCP to sub-cubic HCP. Finally, we provide a procedure to convert any graph to a sub-quartic graph, and using the previous results we conclude that general HCP may be reduced to cubic HCP via a linearly-growing conversion. We conclude with some examples of the savings to be gained by using this approach, compared with the approach in [10].

[1]Manuscripts by Dewdney [6] and Creignou [5] appear to be the primary contributions in this area.

2 Converting Sub-cubic HCP to Cubic HCP

Consider an undirected graph with maximum degree 3. We refer to HCP restricted to such instances as *sub-cubic HCP*. Then there is a simple procedure to convert sub-cubic HCP to cubic HCP.

Sub-cubic HCP to Cubic HCP Conversion Procedure

1. If the sub-cubic instance has any degree 1 vertices, the graph is non-Hamiltonian, and may be replaced by any non-Hamiltonian cubic graph (such as the Petersen graph [15]).

2. Otherwise, replace any degree 2 vertices with a diamond subgraph, as shown in Figure 2.1.

Figure 2.1: A degree 2 vertex, and the corresponding cubic subgraph which replaces it.

It is clear that the resultant graph is cubic. To see that the Hamiltonicity has not been altered, it suffices to recognise that an introduced diamond, once entered, must be fully traversed before departing, as it will be impossible to enter it again. Then this subgraph functions exactly the same as the vertex it replaced, and therefore the new cubic instance contains Hamiltonian cycles if and only if the original instance did.

It is also clear that the above procedure constitutes a linearly-growing conversion. Even in the worst case, where all vertices are of degree 2, the number of vertices is only quadrupled and the number of edges is only sextupled.

It is worth noting that the diamond subgraphs used in the above conversion can be traversed in either of two different ways. Therefore, there is not a 1-1 relationship between the Hamiltonian cycles in the sub-cubic instance, and the Hamiltonian cycles in the converted instance. The 1-many relationship is unavoidable in general, as it is a known result that undirected cubic Hamiltonian graphs must contain at least three Hamiltonian cycles [21], but sub-cubic graphs may contain any number of Hamiltonian cycles, i.e. 1 or 2 in particular.

3 Undirected Graphs with Maximum Degree 4

Consider an undirected graph with maximum degree 4. We refer to HCP restricted to such instances as *sub-quartic HCP*. Then there is a simple procedure to convert sub-quartic HCP to sub-cubic HCP. First we define a subgraph which we call a *4-gate*, as displayed in Figure 3.1. Note that there are four edges, indicated by dashed lines in Figure 3.1, by which the 4-gate may be entered or exited. We will refer to these four edges as *external edges*.

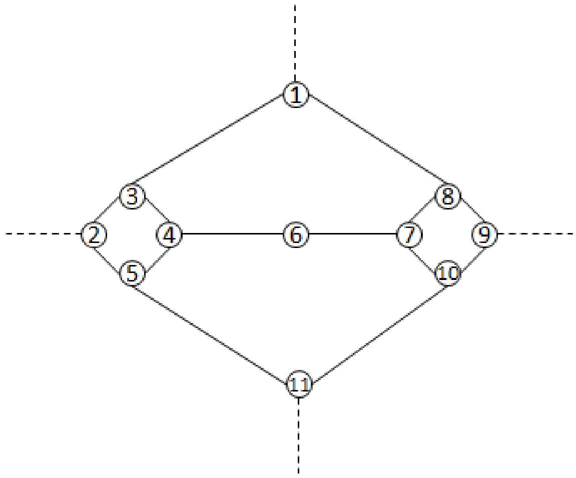

Figure 3.1: A 4-gate, with the dashed lines representing the four external edges.

Lemma 3.1 *It is possible to enter the* 4*-gate via any of the external edges, and exit via any of the remaining external edges, visiting every vertex exactly once.*

Proof: It suffices to give paths between any two of the external edges. Since the 4-gate is undirected, the reverse path is obviously permitted, so we only need to consider unordered pairs. Also, due to symmetry, the top and bottom edges are equivalent, as are the left and right edges. Then there are only three cases that need to be considered, which are displayed in Figure 3.2.

Top edge to left edge: The path is $1-3-4-6-7-8-9-10-11-5-2$.
Top edge to bottom edge: The path is $1-3-2-5-4-6-7-8-9-10-11$.
Left edge to right edge: The path is $2-3-1-8-7-6-4-5-11-10-9$. □

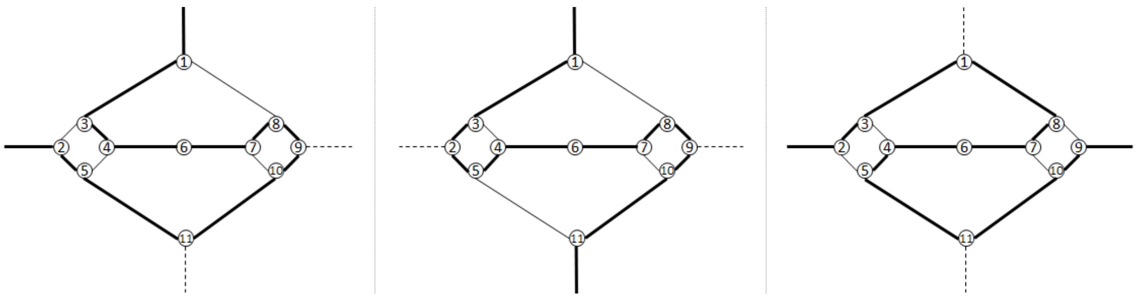

Figure 3.2: The three paths through the 4-gate described in Lemma 3.1, displayed here as bold edges.

In the following proof we make use of the concept of a *live edge*, being one that may still be used without creating a short cycle. In general, if a vertex v has only two live edges, and an adjacent vertex is visited, then vertex v must be visited immediately afterwards in order for a Hamiltonian cycle to be formed.

Proposition 3.2 *Upon entering the* 4*-gate, every vertex must be traversed before exiting in a Hamiltonian cycle.*

Proof: Suppose that during the course of a Hamiltonian cycle, the 4-gate is entered, and then exited before all vertices are visited. Then the Hamiltonian cycle must later enter and exit the 4-gate again. Therefore, one such path must enter or exit through the top edge. Since the graph is undirected, without loss of generality we may assume that the top edge is entered. Then, suppose the cycle travels from vertex 1 to 3. At this point, the cycle may either continue to vertex 2 or 4.

If the cycle continues to vertex 4, it must then continue to the degree 2 vertex 6, and on to vertex 7. Then, since edge $(1, 8)$ was not used, there are only two remaining live edges adjacent to vertex 8, so the cycle must continue to vertex 8 and through to vertex 9. At this point it must exit the right edge (or else it will be impossible to re-enter and re-exit the 4-gate later). Clearly then, since edges $(7, 10)$ and $(9, 10)$ were not used, it is now impossible to visit vertex 10 without getting stuck.

If, instead, the cycle continues to vertex 2, then using the same argument as above, it must exit via the left edge. Then, some time later, the cycle re-enters the 4-gate. Again, without loss of generality, suppose it enters via the right edge. Then using the same argument as above, the cycle is forced to travel the path $9-8-7-6-4-5-11$. At this stage, the cycle must exit via the bottom edge, as it is the only remaining external edge. However, it is then impossible to visit vertex 10. So we conclude that the initial choice of travelling from vertex 1 to 3 is flawed.

However, due to symmetry, travelling from vertex 1 to 8 will be similarly flawed. Therefore the initial assumption that the 4-gate was exited before all vertices was visited must be incorrect. □

From Lemma 3.1 and Proposition 3.2 we see that the 4-gate functions the same as a degree 4 vertex. That is, once it is entered via one edge, any of the other three edges can be departed from, but only once the entire 4-gate has been traversed. Then, the procedure to convert a sub-quartic instance to a sub-cubic instance is as follows.

Sub-quartic HCP to Sub-cubic HCP Conversion Procedure

1. Replace any degree 4 vertices with a 4-gate, with the four adjacent edges to the degree 4 vertex forming the four external edges to the 4-gate.

Since the 4-gate subgraphs are sub-cubic, and all remaining vertices in the original instance are degree 3 or less, the resulting instance is now sub-cubic. It is clear that the conversion from sub-quartic HCP to sub-cubic HCP is a linearly-growing conversion, where in the worst case there are 11 times as many vertices, and 4.5 times as many edges. Then, since there is a linearly-growing conversion from sub-cubic HCP to cubic HCP, we conclude that there is a linearly-growing conversion from sub-quartic HCP to cubic HCP.

The 4-gate is a special case of a more general object which we call an s-gate, displayed in Figure 3.3. Much like for the 4-gate, once the s-gate is entered by a Hamiltonian cycle all vertices must be visited before departing, and it is possible to enter and exit via any pair of external edges. It is clear that all vertices in the s-gate have degree no greater than 3 except for the first vertex, which has degree $s-1$. It is then clear that any graph could be converted to a sub-cubic graph by iteratively replacing any vertices of degree s with s-gates, then any vertices of degree $s-1$ (including those introduced by the s-gates) with $s-1$-gates, and so forth. It is also easy to see that this approach can be generalised to work for directed graphs as well, where a vertex of in-degree r and out-degree s should be replaced by an $(r+s)$-gate. However, this procedure results in a quadratically-growing conversion, and so we do not include it in this manuscript. It should be noted that for graphs of relatively low order and maximum degree, this approach often results in a smaller instance than the linearly-growing conversion in the following section. It should also be noted that if the maximum degree in the original graph is bounded above by a constant, the conversion is linearly-growing.

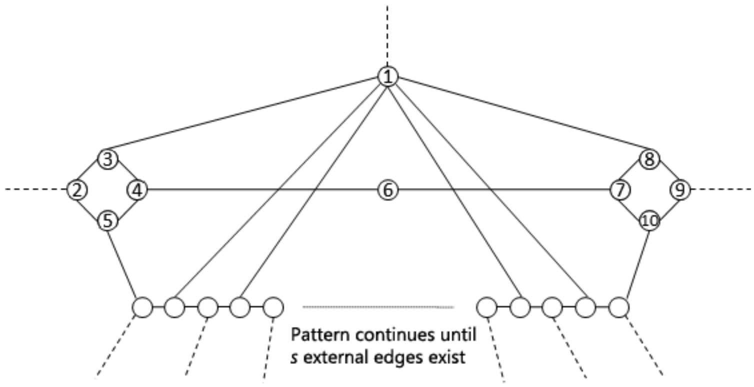

Figure 3.3: An s-gate, with the dashed edges representing the s external edges.

4 A Linearly-growing Conversion from General HCP to Cubic HCP

We now outline a procedure by which any instance of general HCP may be reduced to a sub-quartic instance of HCP. During the conversion we will make use of three special subgraphs, which we call a split, an in-split and an out-split, displayed in Figure 4.1. We refer to the edges on the outside of the subgraphs which connect to the rest of the graph as *external edges*.

Consider a graph containing any of the above three subgraphs. It is trivial to check that any Hamiltonian cycle, upon entering the subgraph, must traverse every vertex before exiting. It is also trivial to check that it is possible to travel from any incoming external edge to any outgoing external edge. Therefore we can replace any vertex by any of the above subgraphs without altering Hamiltonicity. Note that the number of Hamiltonian cycles does not grow when a vertex is replaced by one of the above subgraphs.

We now consider a conversion using the above three subgraphs. Suppose that a general HCP instance is given. It will be assumed that every edge in the graph is directed, and so any undirected edges should be thought of as two individual directed

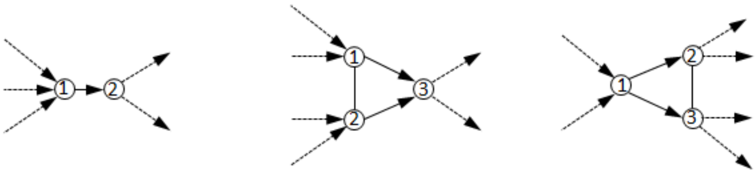

Figure 4.1: A split, an in-split, and an out-split respectively. The arrow-less edge in each of the in-split and the out-split represents two directed edges between the same two vertices, which functions as an undirected edge. The dashed edges represent directed external edges.

edges. The objective will be to replace all vertices of large in-degree or large out-degree (or both) with subgraphs, such that the final graph has maximum in-degree and maximum out-degree below a fixed constant, say d. We will choose $d = 3$, but as will become clear later, d may be chosen as any integer value of 3 or above if so desired.

Consider a particular vertex v in the graph, with in-degree s and out-degree r, where $\max(s, r) > d$. We may replace this vertex with a subgraph by a procedure called the Splitting Procedure, which we outline below. Note that throughout the Splitting Procedure we refer to replacing vertices with the subgraphs described above. This should be done in such a way that the incoming edges adjacent to the replaced vertex form the incoming external edges in the subgraph, and likewise for the outgoing edges. For the in-split, there are two sets of incoming external edges, and so the incoming edges adjacent to the replaced vertex should be shared equally (or different by one, if there is an odd number) between these two sets. Likewise, for the out-split, there should be an equal share of the outgoing edges in each of the two sets of outgoing external edges. An example of such a replacement, using an in-split, is displayed in Figure 4.2.

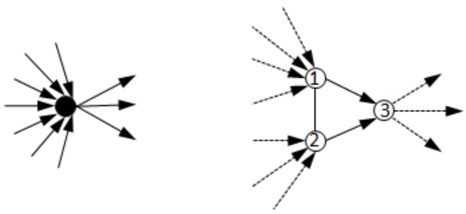

Figure 4.2: A vertex with in-degree 7 and out-degree 3, and the corresponding in-split that replaces it.

Splitting Procedure

1. Replace vertex v with a split.

2. While a vertex in the subgraph has in-degree greater than d, replace that vertex with an in-split.

3. While a vertex in the subgraph has out-degree greater than d, replace that vertex with an out-split.

Lemma 4.1 *The Splitting Procedure terminates in finite time for any $d \geq 3$.*

Proof: Consider a vertex with in-degree s and out-degree r. The first step of the Splitting Procedure occurs only once. After it has concluded, the first vertex in the split will have in-degree s and out-degree 1, and the second vertex will have in-degree 1 and out-degree r.

Next, the second step runs for as long as vertices with in-degree greater than d exist. Each time a vertex with in-degree s is replaced with an in-split, the maximum in-degree of the replacing subgraph is $\lceil \frac{s}{2} \rceil + 1$. It is easy to see that the resultant in-degree will shrink by an integer amount for any $s \geq 4$. Therefore step 2 will conclude in finite time for any $d \geq 3$. The final step is equivalent to the second step, except the arguments involve the out-degree. Therefore, this step will also conclude for $d \geq 3$, and the algorithm will therefore terminate in finite time. □

Since the Splitting Procedure terminates in finite time, and cannot alter the Hamiltonicity of the graph, the Splitting Procedure is guaranteed to convert any vertex of large in-degree or large out-degree (or both) to an equivalent subgraph of maximum in-degree and out-degree of d. Then if this is performed on all vertices in the graph, the resultant graph instance is equivalent to the original graph instance, but has in-degree and out-degree bounded above by d. The only remaining task is to determine how large the resultant graph will be after the Splitting Procedure is applied to all vertices.

Proposition 4.2 *Consider a vertex v with in-degree $s \geq 3$ and out-degree no greater than 2. After the Splitting Procedure is completed for $d = 3$, the resultant subgraph that replaces v has $2s - 5$ vertices.*

Proof: We will prove the result by induction on the value of s. For $s = 3$ the result is trivial. Suppose the result is true for all in-degrees in the interval $3, \ldots, s - 1$. Then, after one iteration of the Splitting Procedure, v is replaced with three vertices whose in-degrees are $\lfloor \frac{s}{2} \rfloor + 1$, $\lceil \frac{s}{2} \rceil + 1$, and 2 respectively. Each of these vertices also has out-degree no larger than 2. Then, by the induction hypothesis, the Splitting Procedure will replace these vertices with $2\lfloor \frac{s}{2} \rfloor - 3$ vertices, $2\lceil \frac{s}{2} \rceil - 3$ vertices, and 1 vertex respectively, which is $2s - 5$ vertices in total, agreeing with the induction hypothesis. Therefore, by induction the result is proved. □

Clearly an equivalent argument to the above can be made for a graph with out-degree $r \geq 3$ and in-degree no greater than 2, which leads to the following corollary.

Corollary 4.3 *Consider a vertex v with in-degree s and out-degree r. After the Splitting Procedure is completed for chosen $d = 3$, the resultant subgraph that replaces v has $O(s + r)$ vertices and $O(s + r)$ edges.*

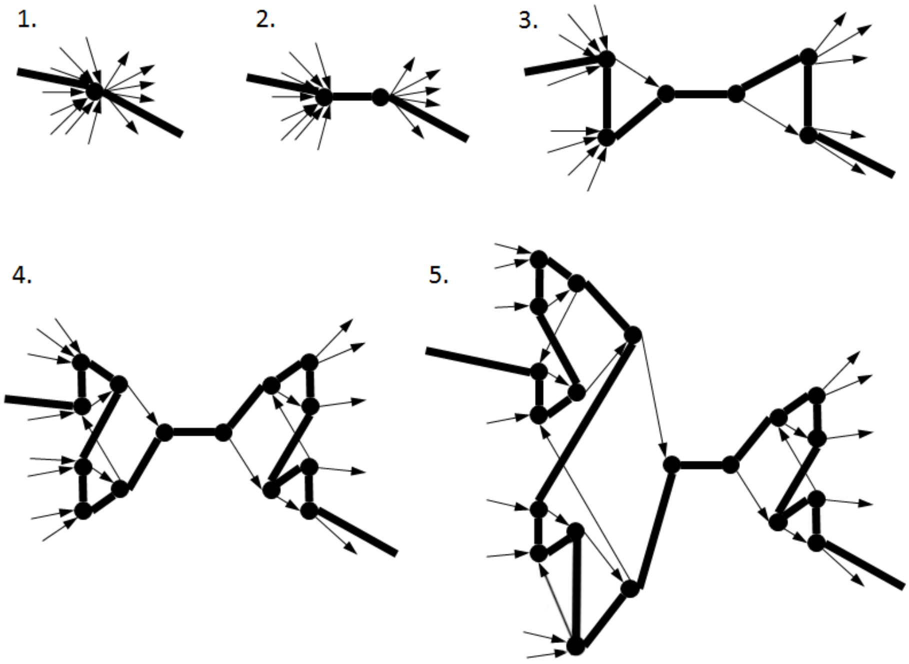

Figure 4.3: A vertex with in-degree 9 and out-degree 6, and the subgraphs produced by applying the splitting procedure for $d = 3$. The bold path shows how the graph HCP would traverse the new graph at each stage.

Proof: Follows immediately from Proposition 4.2 and the fact that all vertices after the splitting procedure have bounded in-degree and out-degree. □

An example of the Splitting Procedure is displayed in Figure 4.3.

It is trivial to see that replacing a vertex with a subgraph, using the above procedure, does not alter the in-degree or out-degree of any other vertex in the graph. Then we may simply perform the above procedure for all vertices in the graph with in-degree or out-degree greater than d. Suppose that, in the original instance containing n vertices, vertex i has maximum in-degree or out-degree d_i, and define $k := \sum_{i=1}^{n} d_i$. It is clear that k at least as big as the number of directed edges in the graph. Then, when the Splitting Procedure is applied to all vertices in the graph, there will be $O(k)$ vertices of in-degree and out-degree no larger than d.

Finally, we present the main theorem of this manuscript.

Theorem 4.4 *Using the procedures described in this manuscript, one may convert any general HCP instance to an equivalent (undirected) cubic HCP instance containing $O(k)$ vertices and $O(k)$ edges, where k is defined as above.*

Proof: Corollary 4.3 demonstrates that it is possible to reduce any instance of general HCP to a directed instance containing a maximum in-degree and maximum out-degree of 3. Then, converting to an undirected graph produces a sub-quartic instance of HCP. Finally, using the procedures outlined in Section 3, an instance of cubic HCP is obtained. All of these procedures describe conversions which are linearly-growing, completing the proof. □

The proof of Theorem 4.4 gives rise to an algorithm for converting any instance of general HCP to cubic HCP, which we call the "HCP to 3HCP Conversion Procedure".

HCP to 3HCP Conversion Procedure

1. Perform the Splitting Procedure on any vertex with in-degree or out-degree greater than 3.
2. Convert the directed graph to an undirected graph.
3. Replace any degree 4 vertices with 4-gates using the Sub-quartic HCP to Sub-cubic HCP Conversion Procedure.
4. Convert the sub-cubic graph to a cubic graph using the Sub-cubic HCP to Cubic HCP Conversion Procedure.

We now conclude this section with an upper bound on the size of the cubic instance obtained from the HCP to 3HCP Conversion Procedure.

Lemma 4.5 *Consider a graph* Γ*, and denote by* k *the sum of in-degrees and out-degrees of all vertices in* Γ*. Then the instance obtained after performing the HCP to* 3*HCP Conversion Procedure will contain no more than* $25k - 60$ *vertices.*

Proof: Consider a single vertex in Γ, with in-degree s and out-degree r. From Proposition 4.2 we know that the subgraph replacing this vertex will contain $2s + 2r - 10$ vertices. Therefore, once step 1 is carried out for all vertices in Γ, the number of vertices in the resultant subgraph will be $2k - 10$. Note that at this stage, there are at most k vertices with maximum in-degree or out-degree of 3 (e.g. see Figure 4.3). In step 2, the number of vertices is tripled, so there are $6k - 30$ vertices. There are still at most k vertices with maximum in-degree or out-degree 4. Exactly $2k - 10$ vertices have degree 2. In step 3, we replace at most k vertices with 4-gates, which each contain 11 vertices, including one vertex of degree 2. So at this point the number of vertices is no more than $16k - 30$, of which at most $3k - 10$ vertices have degree 2. In step 4, we replace at most $3k - 10$ vertices with diamonds, which each contain 4 vertices. So after the HCP to 3HCP Conversion Procedure is completed, the graph contains at most $25k - 60$ vertices. □

For undirected graphs containing m edges, it is trivial to see that $k = 4m$. The following corollary is therefore obvious, and does away with the need to count the in-degree and out-degree of each vertex.

Corollary 4.6 *If* Γ *is an undirected graph containing* m *undirected edges, the cubic graph resulting from the HCP to* 3*HCP Conversion Procedure contains no more than* $100m - 60$ *vertices and* $150m - 90$ *edges.*

5 Improvement over the existing approach

We have implemented the HCP to 3HCP Conversion Procedure using Java and have converted several famous graphs to cubic graphs using our approach. We now compare the corresponding instance input sizes with those of the cubically-growing conversion obtained by first converting from HCP to SAT using the Ariadne100 software package [18], then to 3SAT using the conversion given in Karp [16], and finally to 3HCP using the approach by Garey et al. [10]. Note that, since we do not demand planarity, we may avoid the most expensive step in their approach. In fact, although it is not stated explicitly in [10], it can be relatively easily checked that if the 3SAT instance has m clauses and n literals, the resultant cubic graph will contain $408m+40n$ vertices. It is also interesting to note that Ariadne100 produces SAT instances for which the number of vertices bounds the size, rather than edges. Specifically, if a graph has n vertices and m undirected edges, the SAT instance has $2n^3 - 2n^2 + 2n - 2nm$ and n^2 literals. For completeness, we also include the sizes of instances obtained by iteratively replacing vertices with s-gates, as discussed at the end of Section 3; in many cases this approach outperforms the HCP to 3HCP Conversion Procedure. The comparative results are listed in Table 1, where the savings that may be obtained from using the approaches described in this manuscript are made stark.

Graph	n	**Max, mean Degree**	n **in HCP to 3HCP**	n **in** s**-gate**	n **in Garey**
K_{10}	10	(9,9)	3,560	1,090	845,280
Goldner-Harary [14]	11	(8,4.9091)	1,594	336	1,655,720
Sousselier [1]	16	(5,3.375)	932	96	6,044,160
6-Andrásfai [13]	17	(6, 6)	3,502	782	6,670,664
24-cell [4]	24	(8,8)	7,344	2,064	19,230,336
29-Paley [8]	29	(14,14)	17,574	7,366	30,968,520
Foster Cage [17]	30	(5,5)	4,680	870	41,617,440
Sheehan-40 [20]	40	(39,20.05)	36,316	24,784	80,791,040
Sims-Gewirtz [12]	56	(10,10)	22,736	7,504	271,272,064
Knight's Tour [3]	64	(8,5.25)	10,592	2,416	427,084,800
Sheehan-80 [20]	80	(79,40.025)	152,556	186,324	652,154,880
K_{100}	100	(99,99)	485,600	1,036,900	856,612,800

Table 1: Comparative sizes of cubic graphs obtained from the conversions of some famous higher-degree graphs. The HCP to 3HCP conversion and s-gate conversion are those discussed in this manuscript. The Garey et al. conversion is that which uses (in the final step) the result by Garey et al. [10].

Acknowledgments

The authors gratefully acknowledge useful conversations with Jerzy Filar and Pouya Baniasadi. The research presented in this manuscript was supported in part by the DSTO, and the ARC Discovery Grant DP120100532. We would also like to acknowledge the anonymous referee who made very helpful suggestions that aided the exposition of this manuscript and significantly improved some proofs.

References

[1] R. E. L. Aldred, B. D. McKay and N. C. Wormald, Small Hypohamiltonian Graphs, *J. Combin. Math. Combin. Comput.* 23 (1997), 143–152.

[2] D. W. Barnette, *Conjecture 5*, in W. T. Tutte, *Recent Progress in Combinatorics: Proc. Third Waterloo Conf. Combin., May 1968*, New York: Academic Press, 1969.

[3] A. Conrad, T. Hindrichs, H. Morsy and I. Wegener, Solution of the Knight's Hamiltonian Path Problem on Chessboards, *Discrete Appl. Math.* 50(2) (1994), 125–134.

[4] H. S. M. Coxeter, *Introduction to Geometry, 2nd ed.* New York: Wiley, p. 404, 1969.

[5] N. Creignou, The class of problems that are linearly equivalent to Satisfiability or a uniform method for proving NP-completeness, *Theor. Comput. Sci.* 145 (1995), 111–145.

[6] A. K. Dewdney, Linear transformations between combinatorial problems, *Internat. J. Comput. Math.* 11 (1982), 91–110.

[7] D. Eppstein, The traveling salesman problem for cubic graphs, *J. Graph Algorithms Appl.* 11(1) (2007), 61–81.

[8] P. Erdős and A. Rényi, Asymmetric graphs, *Acta. Math. Acad. Sci. Hung.* 14(3-4) (1963), 295–315.

[9] J.A. Filar, M. Haythorpe and G. T. Nguyen, A conjecture on the prevalence of cubic bridge graphs, *Discuss. Mat. Graph Th.* 30(1) (2010), 175–179.

[10] M. R. Garey, D. S. Johnson and R. E. Tarjan, The planar Hamiltonian circuit problem is NP-complete, *SIAM J. Comput.* 5(4) (1976), 704–714.

[11] H. Gebauer, On the number of Hamilton cycles in bounded degree graphs, in: *Proc. 4th Workshop on Analytic Algorithms and Combin. (ANALCO 08)*, 2008.

[12] A. Gewirtz, The Uniqueness of $g(2, 2, 10, 56)$, *Trans. New York Acad. Sci.* 31 (1969), 656–675.

[13] C. Godsil and G. Royle, The Andrásfai Graphs, in: *Algebraic Graph Theory*, New York: Spinger-Verlag, pp. 118–123, 2001.

[14] A. Goldner and F. Harary, Note on a smallest nonhamiltonian maximal planar graph, *Bull. Malaysian Math. Soc.* 6(1) (1975), 41–42.

[15] D. A. Holton and J. Sheehan, *The Petersen Graph*, Cambridge University Press, 1993.

[16] R. M. Karp, *Reducibility among combinatorial problems*, Springer, New York, 1972.

[17] M. Meringer, Fast Generation of Regular Graphs and Construction of Cages, *J. Graph Theory* 30 (1999), 137–146.

[18] M. Nasu, *Ariadne100: Hamiltonian Circuit Experiment Project*, cited 7th May 2013, `http://www.aya.or.jp/~babalabo/Ariadne/Ariadne.html`, 2000.

[19] G. Royle, *Gordon Royle's Cubic Graphs*, `http://school.maths.uwa.edu.au/~gordon/remote/cubics/`, 1996.

[20] J. Sheehan, Graphs with exactly one hamiltonian circuit, *J. Graph Theory* 1 (1977), 37–43.

[21] W. T. Tutte, On Hamiltonian circuits, *J. London Math. Soc.* 21 (1946), 98–101.

(Received 27 May 2014; revised 19 Sep 2014, 21 Oct 2014)