Improving optimality of n agent envy-free divisions

Stephen W. Nuchia and Sandip Sen

Department of Mathematical & Computer Sciences, The University of Tulsa
600 South College Ave, Tulsa OK 74104, USA
stephen-nuchia@utulsa.edu
sandip@kolkata.mcs.utulsa.edu

Abstract. Division of a resource among multiple agents is a frequent problem in multiagent systems and fair, efficient, and decentralized allocation procedures are highly valued. A division of a good is *envy-free* when every agent believes that its share is not less than anyone else's share by its own estimate. As envy-free procedures are not efficient (in the sense of Pareto optimality) we have previously worked on improving the efficiency of such envy-free division procedures among two agents using models of other agents' utility functions. Though guaranteed envy-free division procedures are available for small number of agents, only approximately envy-free procedures are available for division of a good among an arbitrary number of agents. In this paper, we present an approach by which the outcome of any such approximate algorithms for arbitrary n, or guaranteed algorithms for small number of agents, can be improved in terms of optimality. We propose a two-stage protocol where the first stage identifies possible beneficial exchanges, and the second stage chooses the maximal set of such exchanges that is still envy-free. We map the second stage into a matching problem and present a graph-theoretic algorithm that improves the efficiency of the initial allocations while maintaining the envy-free property.

1 Introduction

A key research issue in agents and multiagent research is to develop negotiation procedures by which agents can efficiently and effectively negotiate solutions to their conflicts [6]. In this paper, we focus on the problem of agents vying for portions of a good. The negotiation process will produce a partition and allocation of the goods among the agents [3, 4]. We are interested in both protocols by which agents interact and appropriate decision procedures to adopt given a particular procedure.

We assume that the good being divided is possibly heterogeneous and the preference of an agent for various parts of the good is represented by a utility function. For historical reasons, we will represent the good as a continuously divisible rectangular piece of cake, where we are only interested in the length of the cake.

In an *envy-free* division, each agent believes that it received, by its own estimate, at least as much as the share received by any other agent [1, 9]. This also implies that an agent has no incentive to trade its share with anyone else. While such guarantees are indeed extremely useful to bring parties to the discussion table, there is no *a priori* reason why a self-interested agent should be happy with just the share that is most

valuable to its estimate in the group when it can possibly get even more. For example, a procedure by which an agent can possibly improve on its share received by an envy-free procedure without losing the guarantee of envy-freeness, would be extremely attractive. Envy-free procedures are also not guaranteed to produce efficient, viz., Pareto-optimal, divisions. This means that it is possible to further re-allocate portions of the cake so that the utility of at least one of the agents is improved without decreasing the utilities of the other agents.

To illustrate this scenario, consider the "divide and choose" procedure, in which one agent cuts the cake into two portions and the other agent gets to choose the portion it wants for itself. The strategy of the cutting agent would be to divide the cake into two portions of equal value by its own estimate. The choosing agent should then choose the portion that is of more value by its own estimate. Thus, both agents would believe they did get the most valuable portion of the cake, and would therefore be envy-free. But there may be portions of the cake in their respective allocations that they can still exchange and further improve their valuations. This problem is only exacerbated with larger number of agents.

In our previous work [7], we presented an approach for two-agent divisions by which agents can use the model of the utility function of the other agent to increase the optimality of the resultant division. The division was guaranteed to be envy-free and was proven to be optimal when such an allocation involved the assignment of a contiguous portion of the cake to one agent. We will refer to this procedure as *2e-opt*.

In the present work, we will focus on the problem of improving the optimality of envy-free divisions among an arbitrary number of n agents. In contrast to our previous work, we will not derive the division from scratch. Instead, we will assume as input an envy-free division of the good among n agents. Such a division with a guaranteed *envy-free* property is know for up to 4 agents; for a higher number of agents, approximate, finite-time procedures have been developed [1]. For our work, we will refer to both the guaranteed and approximate envy-free divisions as envy-free divisions (our discussions will hold for ϵ-envy-free division, where the divisions are envy-free within a bound).

We propose a two-stage reallocation process to improve the optimality of division. In the first stage, every pair of agents divide up their allocated portion of the cake using *2e-opt* and identify beneficial exchanges. In the second stage, a set of such exchanges is chosen that maximizes the total utility value without evoking the envy of any one agent. The procedure presented has the property of maximizing the optimality of the solution starting from the given envy-free division.

To set the context for this work, we will summarize the requisites of fair division processes, the framework for negotiation, and the algorithm *2e-opt* from our previous work. We will then present the two stages of the protocol mentioned above and discuss planned improvements.

2 Division of a good

We concern ourselves with the problem of dividing up a good between multiple individuals. We will assume that the solution procedure is of a decentralized nature. This means that the agents will be required only to abide by a protocol by which the division

is to be made. They can freely choose any strategy to use to determine their actions within the accepted protocol.

For example, a protocol might be that every agent submits a sealed bid for the good. After every bid is collected, the good is divided up among the bidders in proportion to their bids. Our assumption is that once the agents agree to such a protocol, they are free to choose their bids following any strategies they adopt. We require, however, that agents will agree to the division of the good as specified once they have placed their bids.

From a designer's point of view the choice of a protocol provides a platform for agents to negotiate an agreeable division. The choice of a strategy will be dictated by concerns for arriving at a preferred share of the good being divided. The protocol designer should then provide protocols which can be used by agents to successfully negotiate agreeable divisions with reasonable computational costs. We will now look at properties of divisions that can make them agreeable to self-interested agents.

2.1 Desired characteristics of divisions

We assume that a single divisible good is to be divided among n agents. The following criteria have been espoused as desirable characteristics of decision procedures or outcomes from such procedures [1]:

Proportional: Each agent believes that it received, by its own estimate, at least $\frac{1}{n}$ of the goods being allocated.

Envy-free: Each agent believes that it received, by its own estimate, at least as valuable a share as that received by any other agent. This also implies that an agent has no incentive to trade its share with anyone else.

Equitable: A solution, i.e., a partition of the good among the n agents, is equitable, when the share received by each agent is identical in terms of their individual utility functions.

Efficient: A solution is said to be Pareto optimal or efficient if there is no other partition which will improve the perceived share of at least one agent without decreasing the perceived share of any other agent.

2.2 Envy-free divisions with improved efficiency

We assume that the continuously divisible good is possibly heterogeneous and the preference of an agent for various parts of the good is represented by a utility function. Though traditionally represented as a rectangular piece of cake, only the length dimension plays a role in the algorithm. Though we present the *2e-opt* procedure from our previous work [7] here for completeness, at first glance the reader can simply use the results from the following theorems to facilitate understanding of the discussion about the n-person protocol in the later sections of this paper.

The utility to the ith agent of a piece of the cake cut between points a and b (where $a < b$) is given by $\int_a^b U_i(x)dx$, where $U_i(x)$ is the utility function of the ith agent. Without loss of generality, we will assume that the first agent is having a model, $\overline{U_2}$, of

Fig. 1. Our augmentation of Austin's procedure with the modeling agent moving two knives.

the utility function of the second agent. This model need not be accurate, but the better the model, the more benefit the modeling agent stands to gain by using it.

The *2e-opt* procedure we now describe is derived from Austin's moving knife procedure [1]. In the *2e-opt* procedure [7], the modeling agent A holds two knives parallel to the side edges of the cake and then move them to the right allowing for wrap around. Let the positions of the left and right knives at time t be l_t and r_t respectively. l_0 and r_0 determine the initial region offered to B. The knives stop at time $t = T$ when the right knife reaches the original position of the left knife, and the left knife reaches the original position of the right knife. The agent B then chooses a time $\tau \leq T$, and the portion of the cake in between l_τ and r_τ (with wrap-around if needed) is given to B, with the rest of the cake going to agent A (see Figure 1).

It can be shown that B can always produce an envy-free division for itself if it calls "cut" at an appropriate time irrespective of how A moves the knife. B, however, can be resentful as it can presume that the advantage of moving the knife allows A to obtain a super-equitable share for itself, which guarantees a sub-equitable share for B.

We now specify the choice of the initial region, the choice of the target region that the modeler wants the other to choose, and a way of moving the knife such that this target region is most likely to be chosen by the other agent.

Initial region selection: The initial knife placements should satisfy $\int_n^m U_A(x)dx = \frac{1}{2}\int_0^L U_A(x)dx$, where L is the length of the cake. It is also desirable that the region in between does divide the cake in half for agent B. Otherwise B may find it beneficial to choose $t = 0$ (if, according to B's estimate, the region (n, m) is the most valuable) or $t = T$ (if, according to B's estimate, the region (m, n) is the most valuable). But a contiguous region that satisfy both these conditions may not be found in general. So,

the initial placement (m, n) should also satisfy the following condition:

$$(m, n) = \arg\min_{y,x} \left| \int_x^y \overline{U_B}(z)dz - \frac{1}{2} \int_0^L \overline{U_B}(z)dz \right|.$$

For ease of exposition, if $y < x$, we use $\int_x^y f(z)dz$ to represent $\int_x^L f(x)dx + \int_0^y f(z)dz$.

Target region selection: The target region must result in an envy-free division. Eliminating beginning and end points, the target region should be selected from the following set:

$$\begin{aligned} Z = \{(x, y) :(\int_x^y \overline{U_B}(z)dz > max(\int_{l_0}^{r_0} \overline{U_B}(x)dx, \int_{l_T}^{r_T} \overline{U_B}(x)dx)) \\ \wedge(\int_x^y U_A(z)dz < \frac{1}{2} \int_0^L U_A(z)dz)\}. \end{aligned} \tag{1}$$

From the regions in Z, we filter out those regions that are of least value to A: $Y = \arg\min_{(x,y)\in Z} \int_x^y U_A(z)dz$. From the regions in Y, A can select those regions which are estimated to be of most value to B: $X = \arg\max_{(x,y)\in Y} \int_x^y \overline{U_B}(z)dz$. If X is non-empty, then A's goal is to select one of its elements and then move the knives such that the corresponding region is the most attractive offer received by B. If X is empty, however, A will have to be satisfied with half of the cake.

Moving the knife: While moving between point pairs identified in the previous paragraph, the knife locations (u, v) should satisfy the following inequality: $\int_u^v \overline{U_B}(x)dx < \frac{1}{2} \int_0^L \overline{U_B}(x)dx$. For regions where A's utility is high, the spacing between the knives will be reduced to make that region non-envy-free for B.

An instance of the above procedure is shown in Figure 2. We present some properties of this decision procedure (for proof see [7]):

Theorem 1 *Our scheme dominates Austin's procedure with respect to efficiency of allocations.*

Theorem 2 *If a Pareto-optimal allocation for a problem involves a contiguous region, our proposed scheme will select it when given an accurate agent model.*

3 Optimality of n-agent envy-free divisions

Now, we present our two stage protocol for improving the optimality of a given n-person envy-free division of a continuously divisible good. In the first stage of this procedure, each pair of agents identify possible exchanges by dividing each of their shares using *2e-opt*. We define a graph mapping $G = (V, E)$ of this scenario where each vertex $v \in V$ of the graph corresponds to an agent, and the weight w_{ij} on the undirected edge e_{ij} between vertices i and j correspond to the net gain in utilities for a plausible envy-free exchange between agents i and j. An exchange is plausible if both agents gain from it. If there is no plausible exchange between two agents, there is no

Fig. 2. Knife-moving procedure snapshots (G: target region; L,R: left, right knife).

edge between the two corresponding vertices in G (or the corresponding edge weight is zero).

The second stage of the exchange protocol then involves finding the set of such edges with maximum weight such that no agent is envious of the allocations received by other agents after the exchanges, corresponding to the chosen set of edges, are carried out. Any and all agents who are envious of one or more such exchanges can veto the corresponding edges from the chosen set. The protocol consists of first finding a maximal weight edge set such that no more than one edge in the set is incident on any vertex of the graph. This means that an agent will participate in at most one exchange. Next, agents are asked to evaluate the resultant allocations for envy. If the resultant allocation produces envy, the algorithm chooses the next best set of edges. Each iteration of the protocol produces the maximal possible improvement in optimality under the restriction that a single round of pairwise exchanges (identified by *2e-opt*) is executed. The protocol should be iterated until no further improvement is feasible or the available time is exhausted.

In the following we describe these two stages of the protocol in more detail.

3.1 Finding pair-wise envy-free exchanges

In this stage, each pair of agents determine if there is a plausible envy-free exchange between them. We illustrate this process by considering two agents A and B. Let us

Fig. 3. A plausible envy-free exchange between agents A and B.

assume that an envy-free division has produced an allocation of portion X to agent A and portion Y to agent B. Now, *2e-opt* is applied on both X and Y dividing it into portions (X_A, Y_B) and (Y_A, Y_B) respectively. This means that if X were to be the only portion to be divided up between A and B, then (X_A, X_B) would be an envy-free division with A receiving X_A and B receiving X_B, i.e., X_A (X_B) is the more valuable portion of X in A's (B's) estimate. Note that (X_A, X_B) is not the optimal envy-free division possible, but the one that is computed by *2e-opt* (our prior work [7] shows that there is no finite, optimal, envy-free division possible in general). Similar reasoning holds for the portion Y. There exists a plausible exchange between A and B if A believes Y_A is more valuable than X_B, and B believes that X_B is more valuable than Y_A. This process is illustrated in Figure 3. Assuming this exchange is envy-free, the weight in the graph on the edge between A and B will be (A's valuation of Y_A - A's valuation of X_B) + (B's valuation of X_B - B's valuation of Y_A).

This process can be repeated for each pair of agents. Hence in $O(n^2)$ time, where n is the number of agents, we can compute the graph structure mentioned before.

3.2 Choosing optimal envy-free exchanges

In the second stage, the set of edges with maximal weight and satisfying the constraint that no agent participates in more than one exchange and that the corresponding exchanges do not result in envy, has to be chosen. In the following we present a graph-theoretic algorithm for selecting the set of edges of maximal weight such that no agent partakes in multiple exchanges. We also present the procedure for selection of all such edge sets in the decreasing order of their total weight. We traverse down this order as long as the previous selections result in envy. Once a selection is found that is envy-free, we have the best optimality improvement (subject to the use of *2e-opt* in determining plausible exchanges) that is also envy-free given the starting envy-free partition.

The problem of finding improved envy free allocations is related to the maximum weight (vertex) matching problem in the following way. A matching in a simple general graph is a subset of the edges in which no two edges share a vertex. A matching can

be viewed as partitioning the vertices into a set of disjoint pairs and a set of non-paired vertices. The edges in a matching and the vertices adjacent to those edges are said to be *matched*, edges and (especially) vertices that are not matched are said to be *free*. Formally, a matching in a graph $G = (V, E)$ is a subset $M \subseteq E$ such that the maximum degree in the graph (V, M) is one.

The *weight*, $w(M)$, of a matching M is the sum of the weights of the matched edges. A maximum weight matching in G is a matching having the largest weight among all possible matchings in G. A well-known algorithm exists with time complexity of $O(n^4)$ for finding a maximum weight matching in an arbitrary graph [2]. Other more recent algorithms have time complexities of $O(n^3)$ and $O(nm \log n)$, where m is the number of edges and n is the number of vertices in the graph [5]. These algorithms "grow" a matching by identifying what are called *augmenting paths*; the best such path with respect to a matching merged with that matching gives a new, better matching. (The new matching is the *symmetric difference* of the path and the old matching.) The algorithms differ in the details of how a best augmenting path is found.

We have an incremental version of this algorithm that will find a new max weight matching after deleting from the graph an edge found in some max weight matching. In the incremental algorithm the definition of an augmenting path must be relaxed to allow even-length alternating paths and to treat specially paths joining the endpoints of the deleted edge. Using the incremental algorithm is approximately a factor of $\frac{n}{6}$ faster than recomputing the max-weight matching from scratch, though our search works correctly either way. An abbreviated proof of the incremental algorithm given in section A. Analysis of the data structure options for the algorithm is the subject of a future work.

A matching corresponds to a reallocation of the goods and this new allocation may not be envy free. A reallocation is envy free if no participant envies any other participant assuming that the exchanges have taken place. This hypothetical envy is an arbitrary (anti-reflexive) relation on the vertices but, when considering the envy relations defined for two different allocations, the truth value of "u envies v" will not change if the portions allotted to u and v are unaltered.

We search for the independent set of exchanges maximizing total utility without envy using a generate-and-test strategy. Candidate matchings are generated in approximately best-to-worst order and tested in monotonically non-decreasing order. Thus the first matching found to be envy free is a best envy-free matching. The generator is exhaustive but not efficient: it is guaranteed to find all feasible matchings but may visit some more than once.

One further structural property of the envy relation will be useful. The maximum incremental satisfaction available to any participant is a known quantity. Suppose in an instance of the problem some participant k has a potential exchange (an edge) that is not part of the matching being evaluated. It is possible that k would envy (veto) some exchange (edge) based on his current satisfaction but would approve of that exchange if he enjoyed his maximum satisfaction. On the other hand, if some edge would be vetoed by k even when k's best edge was included in the matching then the vetoed edge can not be in any feasible matching for that graph. Incorporating this observation, we now outline the search algorithm.

Begin by finding a best matching M_0 in $G = (V, E)$. Initialize a work queue with the tuple $(\mathcal{P}(E), M_0, w(M_0))$ representing the virtual list of subproblems generated from G — that is, the set of all potentially feasible matchings in G. With the work queue prioritized by the weight of the matchings, largest first, the leading weight is in fact a tight upper bound on the weight of all not-yet-tested matchings.

We now loop until the work queue is exhausted or a feasible matching is discovered. For each tuple $(\mathcal{P}(E_i), M_i, w(M_i))$ extracted we begin by evaluating the envy-free predicate. If the matching is envy free we have found a best envy-free matching and may stop.

If some of the edges in M_i are vetoed we go on to check whether any of those vetoes are permanent in the sense described above. If so we reduce the problem to $(\mathcal{P}(E_i'), M_i', w(M_i'))$ by deleting the permanently vetoed edges from the set representations of E_i and M_i, computing a new max weight matching M_i' in the new subgraph (V, E_i'). Depending on the number of edges deleted, M_i' may be computed incrementally or recomputed from scratch. In any case, the reduced graph contains (or represents) all possible envy-free matchings contained in the original graph. Since in general $w(M_i') \leq w(M_i)$ we insert the reduced tuple into the work queue and loop.

At some point we withdraw an entry from the queue with no permanently infeasible edges. If that matching is nevertheless infeasible (or if we wish to enumerate all feasible matchings) we need to put back into the work queue all the not-yet-tested members of the set of potential matchings represented by that tuple. With M_i a maximum weight matching in the positively weighted edge set E_i, the set

$$\cup_{e \in M_i} \mathcal{P}(E_i - \{e\}) \subseteq \mathcal{P}(E_i) - M_i$$

contains all matchings in E_i except M_i. For every edge e_j in M_i we generate a new work queue entry $(\mathcal{P}(E_i - \{e_j\}), M_{i,j}, w(M_{i,j})$ where $M_{i,j}$ is the maximum weight matching in $(V, E_i - \{e_j\})$ computed incrementally from M_i.

When inserting new entries into the work queue it may be worthwhile to eliminate redundancies in some way. We plan to examine the data structures and time/space tradeoffs associated with various redundancy-reduction strategies.

4 Discussion

In our prior work we presented an algorithm that produced optimal envy-free divisions of continuously divisible goods between two agents under certain assumptions of utility functions. In this paper, we have presented an interesting application of this procedure to improving the optimality of any envy-free division between an arbitrary number of agents. The two-stage protocol we have proposed first identifies all pair-wise envy-free exchanges and then utilizes a graph-based generate-and-test matching algorithm that selects the optimal set of such exchanges that will still produce envy-free allocations.

Our proposed protocol in this paper uses the algorithm presented in our previous work [7] for generating the envy-free exchanges between pairs of agents. The current protocol, however, will work with any algorithm that generates envy-free exchanges between two agents. We have recently worked on a recursive version of our previous algorithm that will be an anytime algorithm to generate the best possible exchange

between two agents given a limit on the number of recursive invocations [8] (the limit is actually the number of cuts allowed on the cake; in the worst case, we will need an infinite number of cuts to generate an optimal exchange, which is not a feasible solution in practice). We can incorporate such a recursive version or any other pairwise exchange mechanism that improves optimality as the first stage of the protocol presented here.

References

1. Steven J. Brams and Alan D. Taylor. *Fair Division: From cake-cutting to dispute resolution*. Cambridge University Press, Cambridge: UK, 1996.
2. J. Edmonds. Maximal matching and a polyhedron with 0,1-vertices. *Journal of Research of the National Bureau of Standards*, B-69:125–130, 1965.
3. Michael N. Huhns and Anuj K. Malhotra. Negotiating for goods and services. *IEEE Internet Computing*, 3(4), 1999.
4. J. Robertson and W. Webb. *Cake Cutting Algorithms*. A. K. Peters, Natick, MA, 1998.
5. Kenneth H. Rosen, editor. *Handbook of Discrete and Combinatorial Mathematics*. CRC Press, Boca Raton, LA, 2000.
6. J. S. Rosenschein and G. Zlotkin. Designing conventions for automated negotiation. *AI Magazine*, pages 29–46, Fall 1994.
7. Sandip Sen and Anish Biswas. More than envy-free. In *Working Notes of the AAAI-99 Workshop on "Negotiation: Settling Conflicts and Identifying Opportunities"*, pages 44–49, 1999. AAAI Technical Report WS-99-12.
8. Sandip Sen and Rajatish Mukherjee. Negotiating efficient envy-free divisions. In *2001 AAAI Fall Symposium on Negotiation Methods for Autonomous COoperative Systems*, pages 149–154, Menlo Park, CA, 2001. AAAI Press. AAAI Tech Report FS-01-03.
9. I. Stewart. Mathematical recreations: Division without envy. *Scientific American*, January 1999.

A Proof of incremental matching algorithm

A condensed proof of our algorithm for finding maximum weight matchings (MWM) after deleting an edge is presented here. An expanded version of this proof is available from the authors.

Let $A \oplus B$ denote the *symmetric difference* $(A \setminus B) \cup (B \setminus A)$ between two sets. We have chosen to use the symbol $\oplus$ to emphasize the isomorphism between set symmetric difference and binary exclusive OR.

A path P in G is *alternating* with respect to a matching M (also in G) if each vertex of degree two in P is adjacent to exactly one edge in M. The *weight* of an alternating path P with respect to a matching M is the sum of the weights of the edges in P when those in M are counted negatively. For an edge set (or graph) E let $w(E)$ be the sum of the edge weights in E. Then the weight of P with respect to M is $w_M(P) = w(P \setminus M) - w(P \cap M)$. Alternating cycles are defined similarly.

A.1 The Difference Component Representation

Let A and B be matchings in a graph G and consider the subgraph induced in G by the symmetric difference $A \oplus B$. We refer to the collection of components of that subgraph[1] as the *component representation* of the difference or DCR and denote it by $\text{DCR}(A, B)$.

Theorem 3 *$\text{DCR}(A, B)$ is a set of paths and cycles in G and is empty only if $A = B$.*

Proof. Any subset of a matching is a matching and $A \oplus B$ is the union of two disjoint subsets of matchings. Each subset can contribute no more than one edge adjacent to any vertex and therefore no vertex in $A \oplus B$ has degree greater than two. □

Let $A \neq B$ and let C be a component of their DCR . Furthermore, let

$$S = \bigcup_{X \in Y} X \quad \text{for some} \quad Y \subseteq \text{DCR}(A, B).$$

Then:

Property 1 *C is alternating with respect to both A and B.*

Property 2 *$A \oplus S$ and $B \oplus S$ are matchings in G.*

Proof. Property 1 is proved using the same reasoning as theorem 3. Proof of property 2 involves a case analysis over the possible degrees of vertices in A, B, and S, all but one case being trivial.

Assume the edge $\{u, v\}$ is in A and the edge $\{v, w\}$, with $u \neq w$, is the only edge in S incident on v. This second edge must be in B since $S \subset A \cup B$ and it cannot be in matching A.

Since the second edge is in B the first edge cannot be and thus both edges are in $A \oplus B$. But then u and w are connected in the subgraph induced by $A \oplus B$ and so the edges connecting them must be in the same component of the DCR . Therefore it is not possible that one is in S and the other is not, contradicting the assumption. □

Note that A and $\text{DCR}(A, B)$ determine a family of matchings, and that family includes B. Let M, M', etc. denote arbitrary matchings in the family determined by A and $\text{DCR}(A, B)$. In particular, let $M' = M \oplus S$.

If the graph is edge weighted then:

Property 3 $w(A \oplus S) = w(A) + w_A(S)$.

Property 4 $w_A(S) = -w_B(S)$.

Theorem 4 *If A and B are maximum weight matchings in G then all components of $\text{DCR}(A, B)$ have zero relative weight:*

$$\forall C \in \text{DCR}(A, B) \quad w_A(C) = w_B(C) = 0.$$

Proof. Assume some component C has non-zero weight. Then by property 4 either $w_A(C) > 0$ or $w_B(C) > 0$ and so

$$w(A \oplus C) > w(A) \quad \text{or} \quad w(B \oplus C) > w(B),$$

contradicting the assumption. □

[1] Strictly speaking, the collection of edge sets of the components.

A.2 Graphs Differing by One Edge

Consider a graph $G = (V, E, w)$ and $e \in E$. Let subgraph $G' = (V, E', w)$ where $E' = E - e$. Assume $w(e)$ is sufficiently large that e is a member of some MWM M in G. Furthermore let M' be an arbitrary MWM in G': necessarily $e \notin M'$.

Theorem 5 $\text{DCR}(M, M')$ *consists of one component* C_e *containing* e *and having non-positive weight relative to* M*. All other components have zero relative weight.*

Proof. We begin by observing that $e \in M \oplus M'$ so C_e must exist. If $w_M(C_e)$ were positive we would have $w(M \oplus C_e) > w(M)$ but the weight of M is assumed to be maximal.

If for some $C \neq C_e$ we have $w_M(C) > 0$ then $w(M \oplus C) > w(M)$, a contradiction.

Because the components of $\text{DCR}(M, M')$ are disjoint $M' \oplus C$ does not contain e; it is a matching in G'. If $w_M(C) < 0$ then by property 4 we have $w_{M'}(C) > 0$ and so $w(M' \oplus C) > w(M')$, another contradiction. □

A.3 The Algorithm

We have reduced the problem of finding an MWM in a graph with a known MWM after removing one edge found in that MWM to the problem of finding an "appropriate" alternating path or cycle containing the edge to be deleted. Rather than directly characterize of the path sought we will describe the search and prove that the result gives us a maximum weight matching.

The singleton path $\{e\}$ is alternating with respect to M so its weight, $w_M(\{e\}) = -w(e)$, is a lower bound on the weight of the path or cycle we seek. Suppose some alternating path or cycle C containing e exists with a larger relative weight: then $C - e$ consists of at most two paths, alternating and having non-negative weight with respect to M.

Search, for example[2] by depth-first exploration, beginning at one endpoint of e and avoiding repeat visits to vertices while alternating edges with respect to M. At each vertex encountered that is not matched in $M - e$ and each time the path length (search tree depth) is even, note the relative weight of the path so far. Prune the tree and separately note the relative weight when the second endpoint of the deleted edge is encountered. If at any time a path weight equal to $w(e)$ is noted the search may be aborted: since $w_M(C_e) \leq 0$ this path weight is maximal.

The result of the first search is a best path P_1 from the first endpoint of e and a best path P_{12} joining[3] the endpoints of e. Repeat the search beginning at the second endpoint of e and avoiding vertices appearing in P_1; let P_2 denote a best path found in the second search. If a path with relative weight equal to $w(e) - w_M(P_1)$ is noted the search may again be stopped early.

Theorem 6 *Let* $\widehat{C} = P_{12} \cup \{e\}$ *if* $w_M(P_{12}) > w_M(P_1 \cup P_2)$ *and* $\widehat{C} = P_1 \cup P_2 \cup \{e\}$ *otherwise. Then* $\widehat{M} = M \oplus \widehat{C}$ *is an* MWM *in* G'*.*

[2] Any suitable optimization from the fast MWM algorithms may be used.

[3] P_{12} will be empty if no alternating cycle in G with respect to M contains e.

Proof. If C_e is a path then $C_e - e$ is a pair of paths (one or both may be empty), one anchored at each endpoint of e. In that case the paths may be of odd or of even length but each will start with an edge not in M. If on the other hand C_e is a cycle then $C_e - e$ is an odd-length path beginning and ending at the vertices of e.

Let $\widehat{M} = M \oplus \widehat{C}$. Since the characteristics of C_e enumerated above are consistent with those of the paths considered by the search algorithm we know $w_M(\widehat{C}) \geq w_M(C_e)$ and therefore $w(\widehat{M}) \geq w(M')$. It remains to show that $\widehat{M}$ is a matching in G'.

First we observe $e \notin \widehat{M} \subseteq E$ so $\widehat{M} \subset E'$.

Because we prune the search if the second endpoint is encountered, we know that the degree of the endpoints of e in $\widehat{C}$ is no more than two. Since edges incident on vertices already visited are not added to the path, no other vertex can have degree more than two. By construction then, $\widehat{C}$ is a path or cycle alternating with respect to M.

The degree of each interior (degree two) vertex in $\widehat{C}$ is unchanged in computing $\widehat{M} = M \oplus \widehat{C}$. Assume $\widehat{C}$ is not a cycle; then two of its vertices have degree one. These vertices correspond to the accepting states of the search algorithm and thus either either have degree zero in M (the odd length case) or are adjacent to an edge that is in both M and $\widehat{C}$ (the even case). In the first case the vertex has degree one in $\widehat{M}$ and in the second its degree there is zero.

All vertices not appearing in $\widehat{C}$ will retain their degrees from M in $\widehat{M}$. Thus $\widehat{M}$ is a matching in G' and since $w(\widehat{M}) \geq w(M')$ it is a maximum weight matching in G'. □