Discrete Mathematics 125 (1994) 279–287
North-Holland

Sharing jugs of wine

Colin J.H. McDiarmid
Corpus Christi College and Department of Statistics, University of Oxford, UK

Jorge Ramirez Alfonsin
Wolfson College and Mathematical Institute, University of Oxford, UK

Received 12 July 1991
Revised 31 March 1992

Abstract

Two people have a full B-gallon jug of wine and two empty jugs of capacity M and S, respectively, where $B = M + S$. Any jug may be poured into any other jug until either the first one is empty or the second is full. What is the quickest way for them to divide the wine equally? We solve this problem and show that a natural related problem may be solved in polynomial time.

1. Introduction

The three-jugs problem. There are three jugs with integral capacities B, M, S, respectively, where $B = M + S$ and $M \geqslant S \geqslant 1$. Any jug may be poured into any other jug until either the first one is empty or the second is full. Initially jug B is full and the other two are empty. (We use B as the name of the jug with capacity B, etc.)

We want to divide the wine equally, so that $\frac{1}{2}B$ gallons are in jugs B and M and jug S is empty, and we want to do so with as few pourings as possible. This problem is a generalisation of the original puzzle with measures $B = 8$, $M = 5$ and $S = 3$. The origin of this puzzle can be traced back at least as far as Tartaglia, an Italian mathematician of the 16th century (see [4] for a historical survey and see also [1, 2, 6]).

We ask three questions. Can we share equally? If so, what is the least number of pourings possible, and how do we achieve this least number? Part of the following theorem may be found in [2].

Theorem 1.1. *It is possible to share equally if and only if B is divisible by $2r$, where $r = \gcd(M, S)$. If this is the case, then the least number of pourings is $(1/r)B - 1$, and the*

Correspondence to: Jorge Ramirez Alfonsin, Mathematical Institute, University of Oxford, 24-29 St. Giles', Oxford OX1 3LB, UK.

0012-365X/94/$07.00
SSDI 0012-365X(92)00206-T

unique optimal sequence of pourings is given by the first $(1/r)B-1$ *steps (pourings) of Algorithm* 1 *below.*

Let us use b, m, s to denote the quantities of wine at any stage in jugs B, M, S, respectively.

Algorithm 1

Pour jug B into jug M
Repeat
Pour jug M into jug S
Pour jug S into jug B
If $m<S$ **then**
Pour jug M into jug S
Pour jug B into jug M

(Note that for simplicity we have not included a stopping condition in this algorithm, but we are interested only in the first $(1/r)B-1$ steps.)

Having just introduced an algorithm, we come to a natural related complexity question (see [3] for terminology and definitions). Consider a three-jugs problem as above. Of course, we need specify only M and S, and define B to be $M+S$. Define a *state* of the problem to be a triple of integers (b, m, s) where $0 \leqslant b \leqslant B$, $0 \leqslant m \leqslant M$, $0 \leqslant s \leqslant S$ and $b+m+s=B$.

THREE JUGS DECISION PROBLEM

Instance: Positive integers M, S and k and two states x, y.
Question: Is it possible to get from state x to state y with at most k pourings?

Theorem 1.2. *The THREE JUGS DECISION PROBLEM is in P (i.e. is solvable in polynomial time).*

There is a natural digraph associated with any three-jugs problem with capacities B, M, S, respectively. Define a *feasible state* to be any state in which at least one of the jugs is empty or full. Observe that there are exactly $2B$ feasible states. (As well as $(B, 0, 0)$ and $(0, M, S)$, for each fixed b with $1<b<B$, there are exactly two feasible states (b, m, s).) It is easy to see that we may restrict our attention to feasible states.

We construct a digraph $D=D(M, S)$ whose vertices are the feasible states, by putting an arc from vertex u to vertex v whenever we can pass directly from u to v with one pouring. This construction is illustrated in Fig. 1. The solution of the puzzle by means of this digraph was first published in [6]. This digraph is central to our investigations. Note, however, that given an instance of the THREE JUGS DECISION PROBLEM, we cannot simply, construct the corresponding digraph and run a shortest path routine. The size of the input is $O(\log B)$ (assuming that $k \leqslant 2B$,

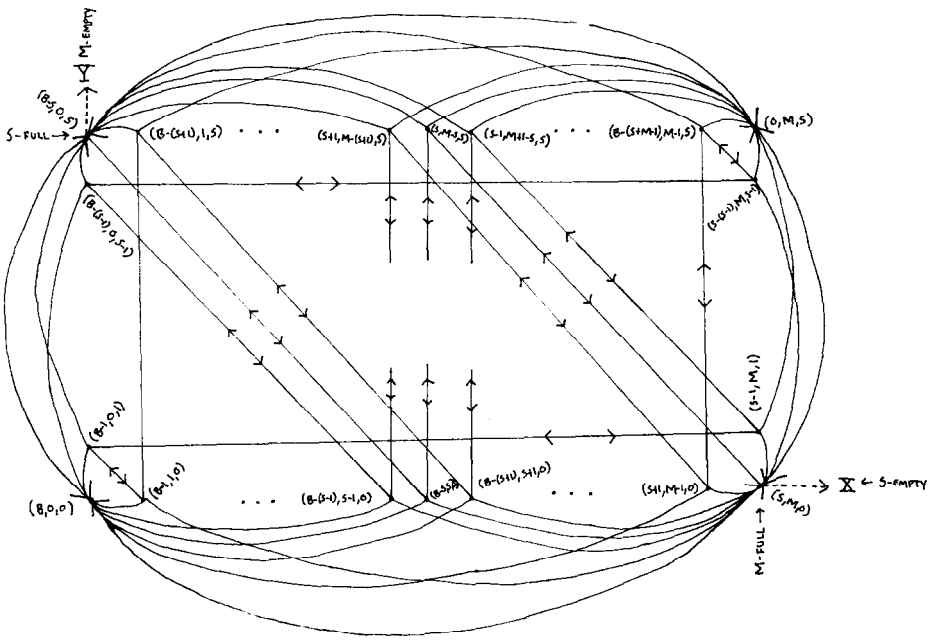

Fig. 1.

say) and a shortest path may have $B-1$ arcs, which is exponentially larger than the input size. The difficulty is that $D(M,S)$ is a large digraph specified rather concisely.

2. The digraph D

We must examine the digraph $D=D(M,S)$ carefully. Assume that $M>S\geqslant 1$ (the case $M=S$ is trivial). Note first that the map $x=(b,m,s)\to\bar{x}=(B-b,M-m,S-s)$ is an automorphism of D, i.e. it is a bijection from the vertex set to itself such that (x,y) is an arc if and only if $(\bar{x},\bar{y})$ is an arc. (Pouring wine from one jug into another is equivalent to 'pouring the empty space' in the other direction.)

We classify the vertices according to the number of coordinates that are 'tight'. We call $(B,0,0)$ and $(0,M,S)$ '3-tight', $(S,M,0)$ and $(M,0,S)$ '2-tight' and the remaining $2B-4$ vertices '1-tight'. The 1-tight vertices $(M,S,0)$ and $(S,M-S,S)$ are 'special'.

The subgraph induced on the 2-tight and 3-tight vertices and the special vertices is shown in Fig. 2 (except for the case $(B,M,S)=(3S,2S,S)$, when there are only six vertices altogether, and the two special vertices are joined by an arc in each direction).

The following statements are all easily checked. For $i=1,2,3$, if a vertex v is i-tight then its outdegree $\delta^+(v)=5-i$. The two 3-tight vertices v have indegree $\delta^-(v)=B$. The

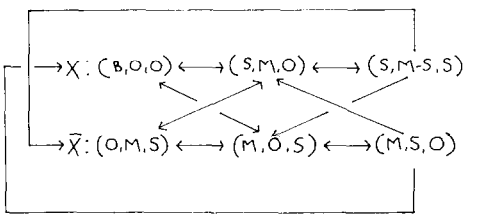

Fig. 2.

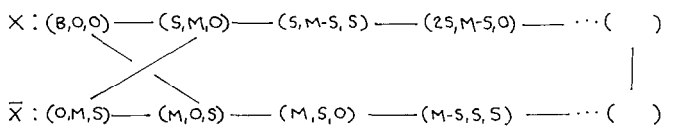

Fig. 3.

two 2-tight vertices v have indegree $\delta^-(v)=B+1$, and the $(2B-4)$ 1-tight vertices v have indegree $\delta^-(v)=2$. Indeed, each nonspecial 1-tight vertex v has an arc to exactly one of the 3-tight vertices and an arc to exactly one of the 2-tight vertices, and an arc (with return arc) to exactly two other 1-tight vertices. The two special 1-tight vertices have one arc to exactly one of the 3-tight vertices, an arc to each of the two 2-tight vertices with one return arc, and an arc (with return arc) to exactly one 1-tight vertex. Finally, note that the only arcs without return arcs are the arcs from the 1-tight vertices to the 2-tight and 3-tight vertices, apart from the two arcs $(S, M-S, S)$ to $(S, M, 0)$ and $(M, S, 0)$ to $(M, 0, S)$ which do have return arcs.

Let G denote the graph on the same vertex set as D, where two vertices are joined by an edge whenever there is an arc each way in D. Then the above discussion shows that each 1-tight vertex has degree 2 and the only 1-tight vertices attached directly to 2-tight vertices in G are the special ones. Thus, the graph G looks as in Fig. 3, if $(M, S) \neq (2, 1)$, together perhaps with some cycle components. We shall see that there are cycle components if and only if $r>1$, where $r=\gcd(M, S)$, and each component of G has exactly $(2/r)B$ vertices.

We shall also want to number the vertices around the 'long cycle' in Fig. 3. This is the cycle from $(B, 0, 0)$ to $(M, 0, S)$ to $(M, S, 0)$, then via 1-tight vertices to $(S, M-S, S)$ to $(S, M, 0)$ and back to $(B, 0, 0)$. In order to do these things, we introduce another algorithm, closely related to Algorithm 1.

3. Algorithm 2 and proof of Theorem 1.1

The following algorithm will show that, when M and S are coprime, the graph G is connected, and it will yield our numbering.

Algorithm 2

Input: Positive integers M, S with $M \geqslant S$ (we set $B = M + S$).

Output: Sequence of states encountered.

Repeat M times

(1) Pour jug B into jug S

(2) Pour jug S into jug M

If jug M is full **then**

(3) Pour jug M into jug B

(4) Pour jug S into jug M

Example. Given $M = 5$ and $S = 3$, the sequence of states encountered by Algorithm 2 is as follows, where we list each state followed by the line number at which the state is reached.

$$(8,0,0),- \quad (5,0,3),1 \quad (5,3,0),2 \quad (2,3,3),1 \quad (2,5,1),2 \quad (7,0,1),3$$

$$(7,1,0),4 \quad (4,1,3),1 \quad (4,4,0),2 \quad (1,4,3),1 \quad (1,5,2),2 \quad (6,0,2),3$$

$$(6,2,0),4 \quad (3,2,3),1 \quad (3,5,0),2 \quad (8,0,0),3 \quad (8,0,0),4$$

Observe that Algorithm 2 traces the long cycle of G in Fig. 3 anticlockwise, whilst Algorithm 1 traces it clockwise. Also, at each execution of line 1 exactly S is poured and at each execution of line 3 exactly M is poured. Thus, a total of MS is poured out of jug B. It follows that jug M is filled exactly S times, and we terminate back in state $(B, 0, 0)$, with the last step being a 'dummy' step.

Partition the vertices $x = (b, m, s)$, other than $(0, M, S)$, into four 'types' as follows. Call x of type 1, if $s = S$; of type 2, if $s = 0$ and $m \geqslant S$ or $m = M$ and $s > 0$; of type 3, if $m = 0$ and $s < S$; and of type 4, if $s = 0$ and $0 < m < S$. The following lemma is easy to check.

Lemma 3.1. *Let x be a vertex (other than $(0, M, S)$) of type t: if Algorithm 2 arrives at vertex x then it does so as at line t.*

Lemma 3.2. *Let M and S be coprime, let $B = M + S$ and let $0 < b < B$. Then there is a unique solution i, j to*

$$B - b = iS - jM, \quad 1 \leqslant i \leqslant M, \ 0 \leqslant j \leqslant S - 1,$$

and it is given by setting

$$iS \equiv B - b \pmod{M} \quad \textit{and} \quad jM \equiv b - B \pmod{S}. \qquad (*)$$

Proof. Any solution must be given by $(*)$. Let i, j be as in $(*)$. Then $iS - JM \equiv B - b \pmod{M}$ and $\pmod{S}$, and hence also $\pmod{MS}$. But

$$iS - jM \leqslant MS < B - b + MS$$

and

$$iS-jM \geqslant S-(S-1)M = M+S-MS > B-b-MS.$$

Hence, indeed, $iS-jM = B-b$, as required. □

Suppose that M and S are coprime, and let $x=(b,m,s)$ be any vertex other than $(B,0,0)$ and $(0,M,S)$. Define i,j as in Lemma 3.2 and let

$$g(x)=\begin{cases} 2i+2j-1 & \text{if } x \text{ is of odd type,} \\ 2i+2j & \text{if } x \text{ is of even type.} \end{cases}$$

Note that $g(x)$ can be computed in time $O(\log B)$ by using the Euclidean algorithm.

Lemma 3.3. *Let M and S be coprime. Algorithm 2 starts at vertex $(B,0,0)$, makes exactly $2B-1$ nontrivial moves arriving exactly once at each vertex other than $(0,M,S)$, then makes one dummy move, and terminates back at $(B,0,0)$. Algorithm 2 does not visit the vertex $(0,M,S)$, and for each vertex x other than $(B,0,0)$ and $(0,M,S)$, Algorithm 2 arrives at x after exactly $g(x)$ steps.*

Proof. Consider a vertex $x=(b,m,s)$, other than $(B,0,0)$ and $(0,M,S)$ (so that $0<b<B$). Define i,j as in Lemma 3.2.

Consider the stage at which Algorithm 2 has just executed line 1 for the ith time. Suppose that lines 3,4 have been executed k times. Then $k>j-2$, since $B-iS+(j-2)M = b-2M < 0$; and $k<j+1$, since $B-(i-1)S+(j+1)M = b+S+M>B$. Thus $k=j$ or $k=j-1$.

(a) Suppose that $k=j$. Then, at the line 1 just executed and the following line 2, the algorithm visits the two distinct vertices with the given value of b.

(b) Suppose that $k=j-1$. Then, at the next 'if test', we have $m=M$ and $s>0$ (since $iS-(j-1)M = B-b+M>M$), so lines 3,4 are executed and the algorithm visits the two distinct vertices with the given value of b.

Thus, for each of the $2B-2$ vertices x other than $(B,0,0)$ and $(0,M,S)$, Algorithm 2 arrives at x after exactly $g(x)$ steps. We already know that Algorithm 2 makes $2B$ steps, with the last step being a 'dummy', and terminates at $(B,0,0)$. The lemma now follows. □

We may now prove Theorem 1.1.

Proof of Theorem 1.1. In the first part of the theorem, the condition that $(1/r)B$ is even is necessary, since all pourings are in multiples of r. From now on, clearly we may assume that $r=1$, and let B be even (see also Lemma 4.1(a)). Observe that $x=(\frac{1}{2}B,\frac{1}{2}B,0)$ is a vertex of type 2 and $\frac{1}{2}B = iS-jM$, where $i=\frac{1}{2}(M+1)$ and $j=\frac{1}{2}(S-1)$. Hence, Algorithm 2 reaches x after $g(x)=2i+2j=B$ steps. Hence,

Algorithm 1 reaches x after $(2B-1)-B=B-1$ steps, and it is clear from Fig. 3 that it follows the unique shortest path. □

4. Parties

Next we consider the case when M and S are not coprime. Assume then that $\gcd(M,S)=r>1$. We partition the $2B$ vertices of the digraph D into r 'parties' as follows. For each $d=0,1,\ldots,r-1$, let the '*d-party*' (denoted by P_d) consist of those vertices (b,m,s) with $m+s\equiv d \pmod r$. Then for each $d\in\{1,2,\ldots,r-1\}$, each vertex in P_d is 1-tight and has arcs in D to exactly one of the two 3-tight vertices ($(B,0,0)$ and $(0,M,S)$) in P_0 and exactly one of the two 2-tight vertices ($(S,M,0)$ and $(M,0,S)$) in P_0. The following lemma is easy to check.

Lemma 4.1. (a) *The map* $x=(b,m,s)\to(1/r)x=((1/r)b,(1/r)m,(1/r)s)$ *gives an isomorphism between the subgraph of the digraph* $D=D(M,S)$ *induced on* 0-party P_0 *and the digraph* $\hat{D}=D((1/r)M,(1/r)S)$.

(b) *Let* $d\in\{1,2,\ldots,r-1\}$. *Given a vertex* $x=(b,m,s)$ *in* P_d, *define*

$$\Psi_d(x)=\begin{cases}(b+d,m-d,s) & \text{if } s=0 \text{ or } s=S,\\ (b+d,m,s-d) & \text{if } m=0 \text{ or } m=M.\end{cases}$$

Then the map Ψ_d *is a bijection between* P_d *and* P_0; $(B-d,0,d)$ *and* $(B-d,d,0)$ *are mapped to the two 3-tight vertices* $(B,0,0)$ *and* $(0,M,S)$; *and for vertices* x,y *in* $\hat{P}_d$, *where* $\hat{P}_d=P_d\setminus\{(B-d,0,d),(B-d,d,0)\}$, x *and* y *are adjacent in the undirected graph* G *(corresponding to* D*) if and only if* $\Psi_d(x)$ *and* $\Psi_d(y)$ *are adjacent in* G.

It follows that, for each $d\in\{1,\ldots,r-1\}$, the subgraph of G induced on P_d is a simple cycle of $(2/r)B$ (undirected) edges, as shown in Fig. 4, which also illustrates the bijection Ψ_d. The original problem looks as in Fig. 5.

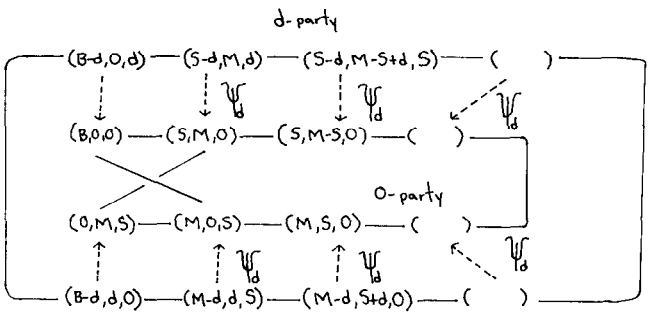

Fig. 4.

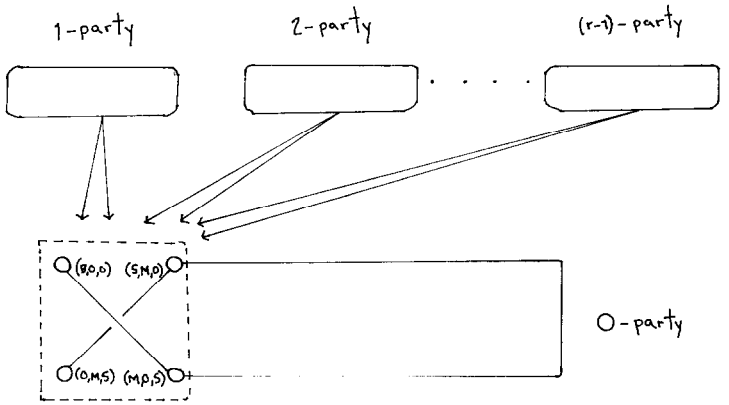

Fig. 5.

5. Finding distances

In this section we shall prove Theorem 1.2. It is easy to see that we can restrict our attention to the case in which x and y are distinct vertices of D (i.e. we can ignore states that are not feasible). Observe that if y is 2- or 3-tight then $d(x,y)$ is 1 or 2, and so can be quickly computed. So we may assume that y is 1-tight. Observe also that $d(x,y)=d(\bar{x},\bar{y})$. Let us consider two cases.

Case 1: $\gcd(M,S)=1$. By Lemma 3.3,

$$d((B,0,0),y)=d((0,M,S),y)=\min\{g(y),2B-1-g(y)\},$$

$$d((S,M,0),y)=\min\{g(y)+1,2B-g(y)-2\}$$

and

$$d((M,0,S),y)=\min\{g(y)-1,2B-g(y)\}.$$

It remains then to consider the subcase when x is a 1-tight vertex. Now there are three possible routes to consider, and we have

$$d(x,y)=\min\{|g(x)-g(y)|,d(x,(M,0,S))+d((M,0,S),y),$$

$$d(x,(S,M,0))+d((S,M,0),y)\}.$$

Case 2: $\gcd(M,S)=r>1$. There are four subcases.

(a) If x and y are both in P_0, then by Lemma 4.1(a) we have $d(x,y)=\hat{d}((1/r)x,(1/r)y)$, where $\hat{d}$ refers to distance in $\hat{D}=D((1/r)M,(1/r)S)$, and we are back to Case 1.

(b) If x is in P_d for some $d\in\{1,\ldots,r-1\}$ and y in P_0, then clearly

$$d(x,y)=\min\{d(x,z)+d(z,y)\colon z \text{ is 2- or 3-tight}\}$$

(and we already know about each $d(x,z)$ and $d(z,y)$).

(c) If x is in P_d and y in $P_{d'}$ for some distinct d, d' in $\{1, 2, \ldots, r-1\}$, then there is no $x-y$ path, since all arcs from P_d go into P_0.

(d) This leaves only the subcase when x and y are both in P_d for some $d \in \{1, \ldots, r-1\}$. Suppose first that x and y are in $\hat{P}_d = P_d \setminus \{(B-d, 0, d), (B-d, d, 0)\}$. Let $\hat{x} = (1/r)\Psi_d(x)$ and $\hat{y} = (1/r)\Psi_d(y)$, and let $t = |\hat{g}(\hat{x}) - \hat{g}(\hat{y})|$ where $\hat{g}$ refers to the digraph $\hat{D}$. Then it follows easily from Lemma 4.1(b) that $d(x, y) = \min\{t, (2/r)B - t\}$. The other possibilities are handled easily from this, using Fig. 4.

6. Related questions

We always assume that $B \geqslant M \geqslant S$. So far, we have considered only the case when $B = M + S$. From what we have already done, it is easy to handle the case $B > M + S$.

Suppose then that $M \geqslant S \geqslant 1$ and $B > M + S$. We start with jug B full and the other two empty, and we want to reach a state with $\frac{1}{2}B$ in jug B. If $B > 2(M+S)$ there is clearly no solution and the case $B = 2(M+S)$ is trivial. Assume now that $B < 2(M+S)$. We may consider the quantity $F = B - (M+S)$ to be 'frozen' in the bottom of jug B. Thus, what we want is the shortest distance in the problem with jug sizes $M+S$, M and S from vertex $(M+S, 0, 0)$ to a vertex with $\frac{1}{2}B - F\ (= M + S - \frac{1}{2}B)$ in the big jug.

The case $B < M + S$ is more complicated and is not pursued here. This case and the corresponding problems with more than three jugs are considered in [5].

References

[1] R. Bellman, L.K. Cook and J.A. Lockett, Algorithms, Graphs and Computers (Academic Press, New York, 1970).

[2] J.A. Bondy and U.S.R. Murty, Graph Theory with Applications (MacMillan, New York, 1975).

[3] M.R. Garey and D.S. Johnson, Computers and Intractibility; A Guide to the Theory of NP-completeness (Freeman, San Francisco, 1979).

[4] T.H. O'Beirne, Puzzles and Paradoxes (Oxford Univ. Press, New York, 1965).

[5] J.R. Alfonsin, Doctoral Thesis, Mathematical Institute, University of Oxford, July 1993.

[6] M.C.K. Tweedie, A graphical method of solving Tartaglian measuring puzzles, Math. Gaz. 23 (1939) 278–282.